



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

Trabajo Fin de Grado

**SISTEMA DE CONTROL DE  
ASISTENCIA DE ALUMNOS A  
TRAVÉS DE TARJETA PERSONAL  
NFC PARA TERMINALES  
ANDROID COMO DISPOSITIVO DE  
CONTROL DE ACCESO**

**Alumno: Alfonso Troyano Montes**

**Tutor:** Prof. D. Juan Carlos Cuevas Martínez

**Tutor:** Prof. D. Ildefonso Ruano Ruano

**Depto.:** Ingeniería de Telecomunicación

**Junio, 2016**

## Índice

1	Memoria	3
1.1	Introducción	3
1.1.1	<i>Descripción del proyecto</i>	3
1.1.2	<i>Objetivos</i>	4
1.1.3	<i>Antecedentes</i>	6
1.1.4	<i>Estado del Arte</i>	6
1.1.5	<i>Tecnologías Utilizadas</i>	7
1.2	Desarrollo	9
1.2.1	<i>Comunicaciones con Docencia Virtual</i>	9
1.2.2	<i>Modelo de Datos</i>	22
1.2.3	<i>Aplicación móvil</i>	28
1.3	Conclusiones	66
1.4	Líneas de Futuro	68
1.5	Anexos	69
1.5.1	<i>Manual de Usuario</i>	69
1.5.2	<i>Manual de Mantenimiento</i>	79
1.6	Bibliografía	89
2	PLANOS	90
3	PLIEGO DE CONDICIONES	91
3.1	Licencias del software	91
3.2	Características del hardware	91
3.2.1	<i>Terminal móvil</i>	91
3.2.2	<i>Tarjetas NFC</i>	92
3.3	Requisitos de los usuarios	92
4	PRESUPUESTO	93
4.1	Diseño del Proyecto	93
4.2	Desarrollo del Proyecto	93
4.3	Entrega del Proyecto	93
4.4	Administración del proyecto	93
4.5	Resumen	94

# 1 MEMORIA

## 1.1 Introducción

En el presente proyecto se describe el diseño e implementación de una aplicación móvil Android para el control de asistencia de alumnos basándose en la tecnología NFC (del inglés *Near Field Communications*) en el móvil y en tarjetas NFC pasivas por parte de los estudiantes. La información de asistencia se gestionará a través de la plataforma de docencia Ilias de la Universidad de Jaén.

Esta aplicación se enmarca en el entorno actual de incremento de potencia y funcionalidad de los teléfonos inteligentes, la aparición de nuevas tecnologías de comunicación y la expansión de plataformas basadas en servicios web, las posibilidades de realizar tareas en modo no presencial se han incrementado de forma exponencial. Se puede acceder desde cualquier punto con suficiente cobertura a las redes personales o profesionales y realizar gestiones en segundos, aumentando la productividad de los usuarios mediante herramientas que además son cada vez más sencillas e intuitivas.

Los nuevos teléfonos inteligentes han aumentado su capacidad de procesamiento, debido a la inclusión de CPUs con varios núcleos, al incremento de memoria RAM y al abaratamiento del precio del almacenamiento sólido, lo cual les permite ejecutar aplicaciones más complejas y potentes. Además, su portabilidad permite imaginar desarrollos más innovadores en los que se pueda aprovechar la ubicación del dispositivo y la conectividad con nuevos periféricos, como chips NFC.

Estas funcionalidades están soportadas por nuevas plataformas basadas en servicios web, que proporcionan una interfaz limpia de cara al desarrollador, ya que éste solamente ha de preocuparse de solicitar una operación y recuperar una respuesta, olvidándose en su mayoría de implementar protocolos complejos de comunicaciones. Estos servicios web pueden incluso estandarizarse, dando lugar a la aparición de paquetes de servicios que pueden ser adquiridos por diferentes empresas y que éstas solamente necesitan configurar internamente para poder aprovechar la vasta cantidad de aplicaciones ya existentes, como por ejemplo la plataforma de docencia virtual basada en Ilias de la Universidad de Jaén.

### 1.1.1 Descripción del proyecto

Este trabajo presenta un sistema de control de asistencia de los alumnos a las sesiones presenciales mediante el uso de la tecnología NFC. Para ello los alumnos harán uso de tarjetas identificativas NFC pasivas que serán leídas por un terminal móvil compatible con esta tecnología y cuya información será procesada por la aplicación Android objeto de este proyecto. Dicha aplicación deberá estar instalada en un dispositivo de control de acceso como un teléfono móvil o tableta, y será la encargada

de almacenar la información correspondiente en un espacio privado ubicado en ILIAS, la plataforma de docencia virtual de la Universidad de Jaén. Para cada asignatura, cuyas sesiones presenciales quieran ser controladas por esta tecnología, deberá existir un espacio específico. La comunicación entre la aplicación Android y la plataforma de docencia se establecerá mediante la tecnología SOAP, los servicios web que implementa ILIAS. La siguiente imagen muestra un esquema de los agentes y tecnologías involucrados en el proceso.

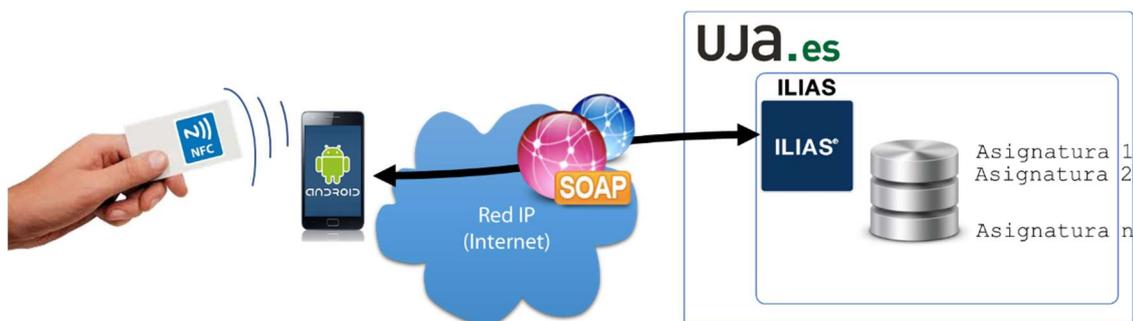


Figura 1 - Esquema del sistema de control de asistencia.

Para ello, la aplicación buscará en dicha plataforma las asignaturas a las que un determinado usuario tiene acceso con privilegios de administración. Una vez localizadas, se descargará la información de los alumnos de dichas asignaturas y permitirá realizar las siguientes operaciones:

- Creación de grupos de prácticas.
- Asignación de los alumnos a los distintos grupos de prácticas.
- Creación de sesiones de prácticas por grupo.
- Control de asistencia a las sesiones, permitiendo a los alumnos el uso de tarjetas personales NFC.
- Carga de los datos de la sesión a la asignatura correspondiente de la docencia virtual para la posterior evaluación de la asistencia.

### 1.1.2 Objetivos

Este proyecto tiene como objetivo el conseguir un control rápido, eficaz y seguro de la asistencia de los alumnos a las sesiones en pequeño grupo, no solamente en las dependencias de la UJA, sino en cualquier otro lugar donde éstas se desarrollen.

Se toma como punto de partida que el profesor es quien dispone del terminal móvil conectado a Internet, el cual dispone de conectividad NFC para poder identificar la tarjeta de alumno.

#### Objetivos principales:

- La información se debe guardar en Ilias en un formato fácilmente editable una vez se ha terminado la sesión, con registro de las entradas y salidas.

- Para ello se va a optar por un formato de hoja de cálculo en texto plano, conocido como csv (del inglés *comma-separated values*), que aúna poco espacio para una transferencia ligera y la posibilidad de ser editado directamente con hojas de cálculo.
- Se optará porque el tutor pueda cargar los datos de la sesión cuando considere.
- Hacer una precarga de los datos de los alumnos de una asignatura antes de la sesión (Datos: nombre completo y fotografía del alumno).
  - Para poder identificar al alumno y contrastar su identidad frente a la tarjeta NFC que se utilice, será necesario cargar las fotos de los alumnos y mostrarlas durante la sesión de prácticas.
- Toma de datos offline-actualización online.
  - Los datos de los alumnos y las asignaturas se tomarán vía online, a través de docencia virtual.
- Seguridad. Se deberán observar las medidas de seguridad pertinentes para evitar casos de suplantación de personalidad, copia de tarjetas, intrusión en las comunicaciones, etc.
  - Se requerirá de un usuario de Ilias habilitado con privilegios de administración de las asignaturas para poder acceder a los datos de las mismas.
  - Se utilizarán los servicios web de Ilias para comunicarse con la plataforma virtual, ya que estos implementan la seguridad necesaria.
  - El tutor podrá comprobar la identidad del alumno in situ, contrastando la información de la tarjeta con la que tiene en pantalla.
- Los alumnos se identificarán a través de una tarjeta NFC personal.
  - La aplicación debe ser capaz de leer tarjetas NFC y gestionar las entradas y salidas de los alumnos en la sesión de prácticas actual gracias a ella; pero además debe permitir la edición manual de entradas y salidas por parte del tutor, en caso de utilizar un terminal sin NFC o que el alumno no disponga de su tarjeta identificativa.

**Objetivos opcionales:**

- Uso de la emulación de Contacless Card a través de NFC en Android.
- Control de alumnos que realizan sesiones de prácticas en grupos distintos al que pertenecen.
  - Se ideará un sistema que permita a alumnos recuperar sesiones de prácticas de forma transparente al tutor.

- Se utilizará una estrategia que permita a los alumnos no perder sesiones de prácticas en caso de que los grupos no estén decididos.

### 1.1.3 Antecedentes

El presente trabajo se basa en el desarrollado por José Manuel Jiménez Bravo, denominado “Aplicación para el control de acceso a dependencias y edificios basado en NFC (Near Field Communication) para terminales Android”, el cual ha supuesto un buen punto de partida para esta aplicación, ya que se puede reutilizar parte de la funcionalidad implementada para manejar los servicios web de Ilias y realizar lecturas NFC.

En ese proyecto se desarrolló una aplicación capaz de realizar un control de asistencia a prácticas, utilizando etiquetas NFC ubicadas en las dependencias donde se celebra la sesión. Se utilizan dos etiquetas por dependencia: entrada y salida, las cuales son leídas por los dispositivos de los alumnos al acercarlos a ellas. Esta acción, dispara un evento que ejecuta la aplicación objeto de ese proyecto en dicho dispositivo, y, leyendo el tipo de etiqueta a la que se ha acercado el mismo, genera un registro de asistencia y lo carga en la plataforma de docencia virtual.

### 1.1.4 Estado del Arte

Actualmente disponemos de terminales Android, ampliamente extendidos, bajo su versión 6.0, que, como novedad destacada, son capaces de realizar identificación del usuario mediante lectura de huella dactilar de forma nativa; aunque la funcionalidad NFC data desde la versión 2.3.

La tecnología NFC (Near Field Communication), es una tecnología de transmisión punto a punto, inalámbrica, pero de corto alcance (unos 20 cm.), que opera en la banda de los 13.56 MHz, soportando una tasa de transferencia de 424 Kbps y utilizando SSL para garantizar la seguridad de la comunicación. Está basada en la capacidad de generar un pequeño campo magnético por, al menos, una de las entidades que forman la comunicación. Atendiendo a esto, se definen dos modos de funcionamiento:

- Modo Pasivo: cuando solamente una de las entidades es capaz de generar el campo magnético necesario para la comunicación. En este caso el terminal que no es capaz de generar dicho campo se alimenta del mismo para realizar comunicaciones en modo semi-duplex con el otro. Se utiliza para el intercambio de pequeñas cantidades de datos, con utilidades como la publicidad o la identificación personal.
- Modo Activo: cuando las dos entidades de la comunicación pueden generar el campo electromagnético. Funciona en modo full-duplex, por lo que se ofrecen más posibilidades de comunicación que en el modo pasivo, ya que, en éste, ambos elementos pueden implementar aplicaciones complejas para

realizar operaciones, tales como emparejamiento de dispositivos, pagos electrónicos, intercambio de datos, ...

Sus principales usos son:

- Validar equipos y personas: acercando un teléfono móvil o una tarjeta provista de chip NFC a un dispositivo de lectura, se puede identificar al portador, tal y como se va a utilizar en este proyecto y como se hizo en el proyecto mencionado en *Antecedentes*.
- Intercambiar datos: se puede transferir información entre entidades provistas de NFC con un simple contacto. Al activarse el campo magnético NFC se puede iniciar una transferencia de datos, que típicamente es de pequeño volumen. Valga como ejemplo la posibilidad de configurar un terminal Android a partir de uno anterior, en la que es posible comunicarle al nuevo dispositivo la cuenta a utilizar y, a partir de ahí, el otro dispositivo comienza su propio proceso de configuración.
- Realizar pagos a través del dispositivo móvil: al acercar el dispositivo móvil a un terminal de pago NFC se puede realizar automáticamente un cobro, si dicho dispositivo tiene asociada una tarjeta de crédito o permite cargos a través del operador de telefonía, tal y como se permite en aplicaciones como Android Pay, Apple Pay o Vodafone Wallet.

Existe una creciente variedad de aplicaciones de control de presencia utilizando tecnología NFC, destacando la que se puede contratar en [www.controldehoras.com](http://www.controldehoras.com), con un funcionamiento muy similar a la aplicación objeto de este proyecto, aunque con la diferencia de que carece de especialización en la necesidad específica de gestionar grupos de prácticas durante el curso de una asignatura y sin comunicación directa con la plataforma de docencia virtual de la Universidad de Jaén.

### 1.1.5 Tecnologías Utilizadas

Como entorno de desarrollo se dispone de Android Studio 2.0, que es la plataforma oficial para crear aplicaciones basadas en Android y que goza del mayor soporte de Google en la actualidad. Dicho entorno incorpora como novedad principal la posibilidad de editar código y continuar el proceso de depuración, lo cual agiliza la fase de pruebas de los desarrollos.

Android Studio soporta el desarrollo de código mediante el lenguaje Java, que es el único para el que Google, responsable de la plataforma Android, da soporte de manera oficial. Java es un lenguaje de programación de propósito general, concurrente y orientado a objetos, que destaca por ser multiplataforma sin necesidad de volver a compilar las aplicaciones, mantenido actualmente de Oracle.

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares abiertos que sirven para intercambiar datos entre aplicaciones. Esto significa que distintas aplicaciones, desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma pueden utilizar estos servicios para intercambiar datos entre ellas. Las comunicaciones mediante servicios web necesitan, por un lado, de una entidad proveedora de servicios web como la plataforma de docencia virtual basada en Ilias, de un consumidor de servicios como la aplicación objeto de este proyecto y de un descriptor de servicios web, que permita al consumidor conocer los servicios existentes y sus parámetros. Este descriptor es un fichero XML (del inglés *eXtended Markup Language*), basado en un esquema WSDL (*Web Services Description Language*). Existe un protocolo, denominado UDDI (del inglés *Universal Description, Discovery, and Integration*), para comunicar los servicios existentes y los argumentos para operar con ellos. En proveedores de servicios de ámbito local, UDDI no se suele utilizar, con el objetivo de aminorar costes y las aplicaciones se conectan a una URL conocida del proveedor para descubrir su WSDL, como es el caso de la Universidad de Jaén.

## 1.2 Desarrollo

Para una comprensión más completa de la aplicación, se va a exponer el proceso de desarrollo seguido, junto con las dificultades acontecidas y las distintas soluciones adoptadas.

### 1.2.1 Comunicaciones con Docencia Virtual

Para empezar, hay que analizar qué operaciones de comunicación se necesitan con la plataforma de docencia virtual para llevar a cabo los objetivos y cómo van a realizarse. Esta plataforma permite operar a través de navegador web y a través de servicios web.

La solución de navegador web consistiría en implementar dentro de la aplicación un navegador de internet y que el usuario fuera realizando las operaciones necesarias para manejar los datos de la sesión y los ficheros. Esta es la solución más sencilla ya que la aplicación podría desentenderse de las comunicaciones con la plataforma de docencia virtual. Sin embargo, esto implicaría por un lado que la aplicación perdería el control de los mensajes de error y por otro, como ya se ha dicho, que el mismo usuario tendría que realizar las operaciones de manera manual y con un interfaz distinto al de la aplicación, lo cual puede resultar confuso y contraproducente en la experiencia de uso.

Por otro lado, al basar la solución en los servicios web se gana en versatilidad y facilidad, ya que no hay que generar las peticiones ni las respuestas y se mantiene el control del proceso de comunicaciones a través de actividades generadas específicamente para la aplicación, lo cual facilitará la gestión de las mismas y la futura la tarea de mantener el código ya que se podrán repetir con sencillez las situaciones conflictivas. Además, se utilizará un mismo interfaz de usuario, lo que redundará en una mejor experiencia de usuario.

Debido a lo expuesto anteriormente, se elige la solución basada en servicios web.

#### 1.2.1.1 Servicios Web

Para empezar, el proveedor de los servicios web debe generar un documento en lenguaje WSDL (Web Services Description Language) en el que se identifiquen todos y cada uno de los servicios web disponibles en la plataforma y cómo se puede operar con ellos. Dicho documento tiene que ser publicado, haciendo uso del protocolo UDDI, para que los clientes puedan descubrir esa información. En el caso de la Universidad de Jaén, la información de los servicios web en formato WSDL es accesible a través de la URL:

- <http://dv.ujaen.es/docencia/webservice/soap/server.php?wsdl>

```

+<types></types>
- <message name="loginRequest">
  <part name="client" type="xsd:string"/>
  <part name="username" type="xsd:string"/>
  <part name="password" type="xsd:string"/>
</message>
- <message name="loginResponse">
  <part name="sid" type="xsd:string"/>
</message>
- <message name="loginCASRequest">
  <part name="client" type="xsd:string"/>
  <part name="PT" type="xsd:string"/>
  <part name="user" type="xsd:string"/>
</message>
- <message name="loginCASResponse">
  <part name="sid" type="xsd:string"/>
</message>
- <message name="loginL.DAPRequest">
  <part name="client" type="xsd:string"/>
  <part name="username" type="xsd:string"/>
  <part name="password" type="xsd:string"/>

```

Figura 2 - WSDL Universidad de Jaén

Se puede consultar en un formato más legible en la URL:

- <http://dv.ujaen.es/docencia/webservice/soap/server.php>

En la que por cada servicio web disponible se mostraría la siguiente información:

```

Name: login
Binding: ILIASSoapWebserviceBinding
Endpoint: http://dv.ujaen.es/docencia/webservice/soap/server.php
SoapAction: urn:ilUserAdministration#login
Style: rpc
Input:
  use: encoded
  namespace: urn:ilUserAdministration
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
  message: loginRequest
  parts:
    client: xsd:string
    username: xsd:string
    password: xsd:string
Output:
  use: encoded
  namespace: urn:ilUserAdministration
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
  message: loginResponse
  parts:
    sid: xsd:string
Namespace: urn:ilUserAdministration
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: ILIAS login function

```

Figura 3 - Información Servicio Web

Pueden apreciarse los puntos más importantes de cara al servicio:

- Name: Nombre del servicio.
- Endpoint: Dirección donde encontrar el servicio.
- Input: Parámetros de entrada que debe recibir el servicio.
- Output: Parámetros de salida que devuelve el servicio.

- Namespace: Espacio de nombres que define el grupo al que pertenece el servicio web a utilizar. Si los permisos del usuario no permiten acceder al namespace, el servicio no se ejecutará.
- Transport: protocolo de comunicaciones. Todos los servicios web de la Universidad de Jaén soportan SOAP.
- Documentation: Resumen de las funciones de este servicio.

Analizando los objetivos que debe alcanzar la aplicación y el documento con la descripción de los servicios, podemos deducir que, a priori, las operaciones a realizar en Ilias y sus servicios web asociados serían:

Acción	Servicio Web
Acceso a la plataforma	<u>Login()</u> : se identifica en la plataforma y obtiene un identificador de sesión que se necesita en las siguientes consultas.
Descubrir asignaturas del tutor	<u>getUserXml()</u> : obtiene los identificadores de sus asignaturas asociadas.
Obtener datos de las asignaturas (nombre, alumnos)	<u>getCourseXml()</u> : obtiene un descriptor con el nombre de la asignatura y sus miembros.
Obtener datos de los alumnos (nombre, DNI, fotografía, ...)	<u>getUser()</u> : obtiene los datos del usuario
Subir ficheros csv con los datos de asistencia	<u>addFile()</u> : añade un fichero a la plataforma.

*Tabla 1 - Propuesta de servicios web*

Al empezar a trabajar con los servicios web en un primer acercamiento, se observa que, para solicitar la información asociada a un usuario, no es válido indicar el identificador con el que dicho usuario hace login en la plataforma, sino que hay que descubrir el identificador interno que le da Ilias. Esto se hace a través del servicio **getUserIdBySid()**, que devuelve esta información a partir del parámetro retornado de la operación **login()**.

Una vez resuelto este problema, se descubre que no es posible obtener los datos de ningún usuario (ni siquiera uno mismo) a través del servicio web **getUser()**, debido a la propia configuración de Ilias en la Universidad de Jaén ya que siempre se obtiene el mensaje "Check Access Failed", indicando falta de privilegios, en una respuesta como la siguiente:

```
<SOAP-ENV:Body>
```

```

<SOAP-ENV:Fault>
  <faultcode xsi:type="xsd:string">Server</faultcode>
  <faultstring xsi:type="xsd:string">Check access failed.</faultstring>
  <faultactor xsi:type="xsd:string"/>
  <detail xsi:type="xsd:string"/>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>

```

No existe ningún otro servicio con las capacidades de **getUser()** para obtener los datos de los alumnos, por lo que mientras se mantenga la configuración de Ilias no va a ser posible operar de esta manera.

Esto obliga a adoptar una estrategia diferente para descubrir cuáles son los alumnos de cada asignatura. Se opta, como medida alternativa, porque cada asignatura contenga un fichero en formato de texto separado por comas, con los alumnos matriculados en la misma, con el nombre de alumnos.csv.

Este fichero se puede generar automáticamente por Ilias; pero hay una dificultad añadida: Ilias no incluye en el fichero la fotografía del alumno, por lo que la funcionalidad de identificar al alumno a través de su imagen asociada quedaría descartada mientras se usara esta modalidad.

Para poder recuperar dicho fichero hay que hacer uso de dos servicios web, ya que hay que realizar dos acciones:

Acción	Servicio Web
Descubrir el identificador del fichero	<u><b>getObjectsByTitle()</b></u>
Descargar el fichero	<u><b>getFileXML()</b></u>

*Tabla 2 - Servicios Web para descarga de ficheros*

Analizando la respuesta del servicio web **getObjectsByTitle()**, se puede observar que además del identificador interno del fichero, también se obtiene el identificador del directorio en el que éste está almacenado y el de la asignatura propietaria. Esto, unido a la configuración de permisos de Ilias, que solamente devuelve los ficheros a los que el usuario tiene suficientes privilegios; permitiría descubrir las asignaturas asociadas a un tutor, los identificadores de sus ficheros de alumnos y el directorio en el que almacenar los resultados de asistencia con una misma operación, quedando el protocolo de comunicaciones de esta forma (los parámetros de entrada y salida se analizarán más adelante):

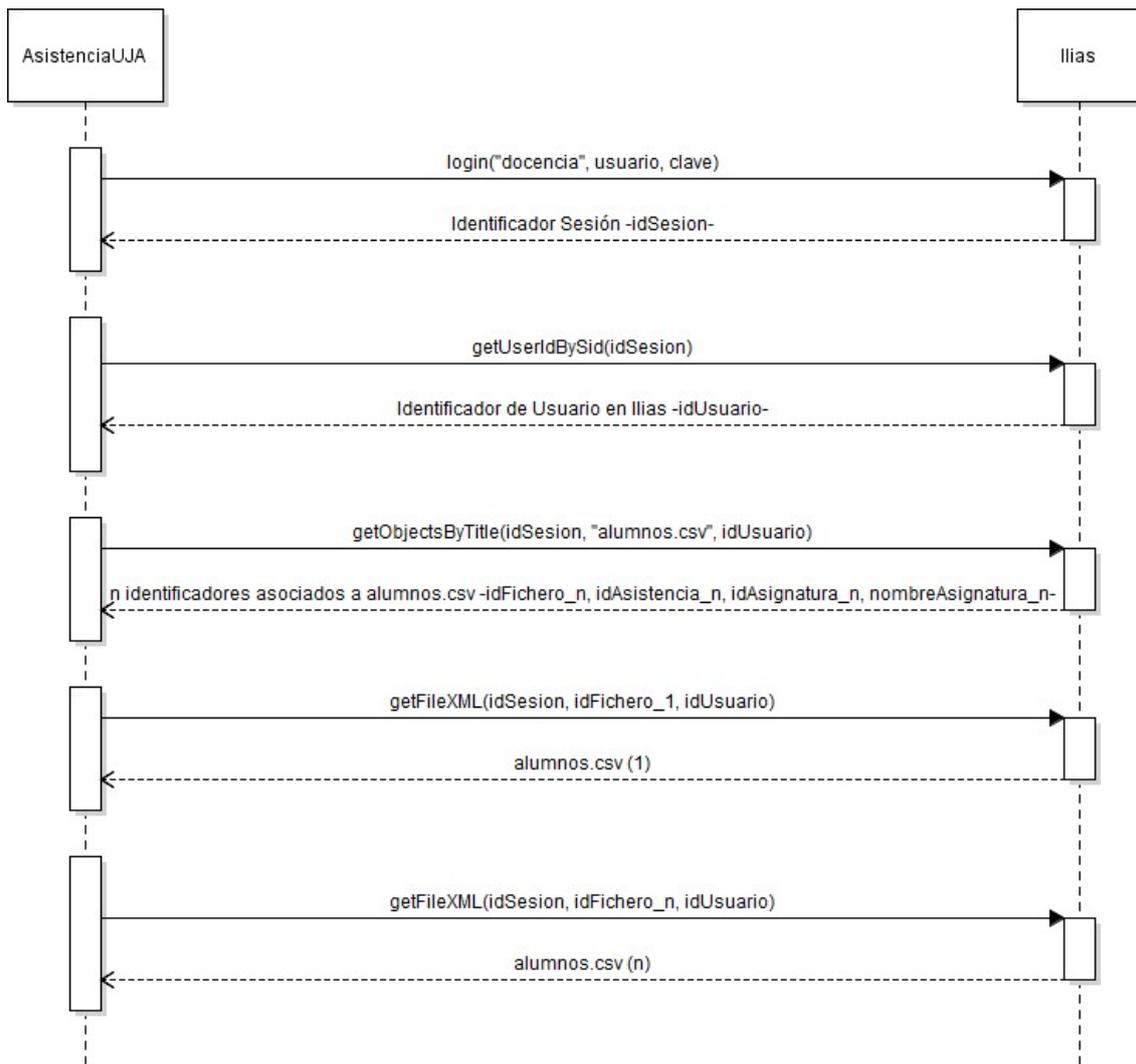


Figura 4 - Protocolo comunicaciones login - asignaturas - alumnos

Por otro lado, a la hora de almacenar los datos de asistencia en Ilias se nos plantean tres opciones:

- Un fichero por sesión.
- Un fichero por grupo de prácticas.
- Un fichero por asignatura.

La primera opción es descartada debido a la inviabilidad a la hora de manejar tal cantidad de ficheros y a la poca usabilidad final para contabilizar las asistencias.

La segunda opción tiene como punto a favor que se pueden analizar patrones en la asistencia de los alumnos por grupo y como puntos en contra que además de la operación de subir fichero, hay que considerar la de actualizar fichero (servicio web **updateFile()**) y que el tutor tendría que gestionar varios ficheros por alumno para hacer un cómputo global, ya que los alumnos que recuperaran horas en otros grupos, estarían en el fichero de ese otro grupo.

La tercera opción tiene como ventaja principal que el tutor solamente necesitaría trabajar con un fichero para obtener los datos de asistencia de cada alumno. Además, se pueden analizar patrones de asistencia y recuperación de horas si se le da al fichero un contenido adecuado y como punto en contra que también se necesitaría utilizar el servicio web de actualización de ficheros.

Poniendo todas las opciones sobre la mesa, se decide generar un único fichero por asignatura, ya que proporciona más ventajas al tutor que un fichero por grupo (y que por sesión), con los mismos inconvenientes.

El fichero de asistencia será creado en Ilias de manera automática la primera vez que se vayan a cargar datos de asistencia a la plataforma de docencia virtual, bajo el nombre AsistenciaPracticas.csv y contendrá únicamente la línea de cabecera del fichero (que será definida en el apartado 1.2.2.2). Cada vez que se añadan datos al mismo, se descargará todo el contenido del fichero mediante el servicio **getFileXML()** para añadir la nueva información de asistencia al final. Seguidamente se utilizará el servicio web de actualización de ficheros para cargar todo el contenido a la misma referencia Ilias, ya que dicho servicio reemplaza el nuevo contenido por el que ya existiera en la plataforma.

El protocolo de comunicaciones sería:

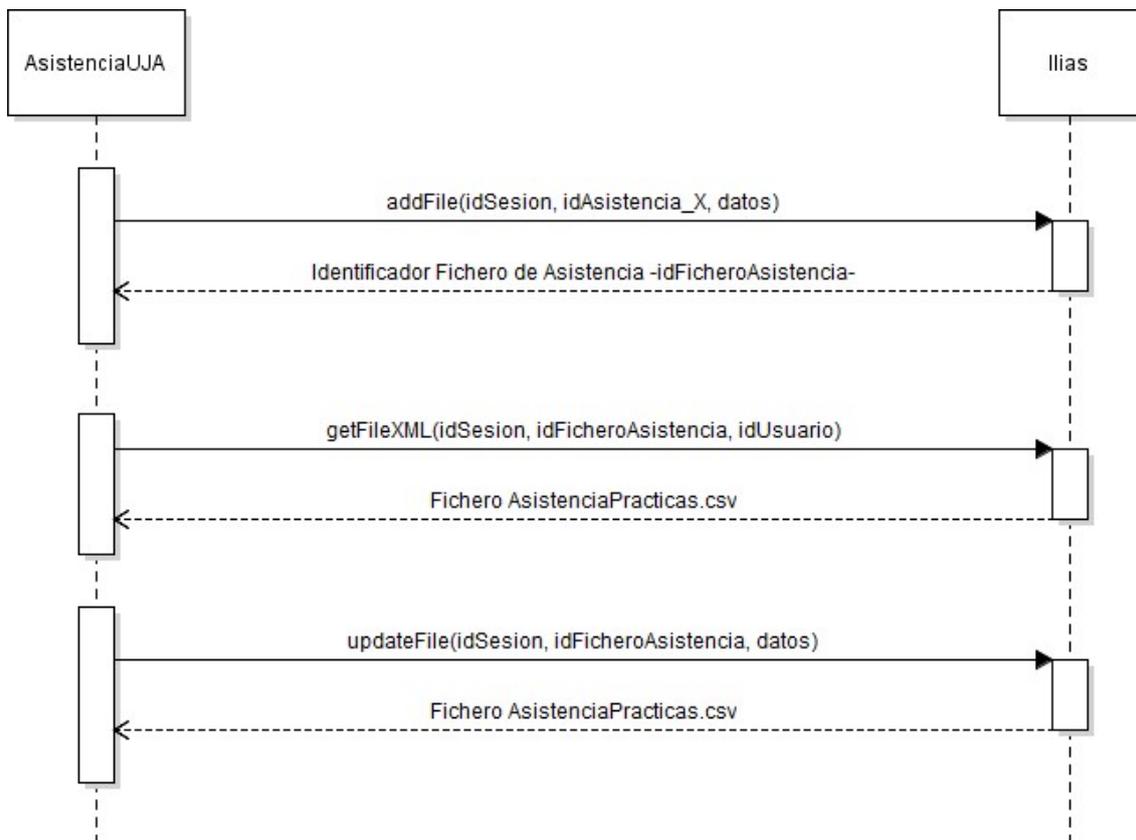


Figura 5 - Protocolo comunicaciones añadir y actualizar fichero asistencia

Por tanto, los servicios web a utilizar serán los siguientes:

Acción	Servicio Web
Acceso a la plataforma	<u>Login()</u> : se identifica en la plataforma y obtiene un identificador de sesión que se necesita en las siguientes consultas. <u>getUserIdBySid()</u> : obtiene el identificador interno del usuario que inicia sesión, para comprobar sus permisos en las siguientes peticiones.
Descubrir asignaturas del tutor Obtener datos de las asignaturas (nombre, alumnos) Obtener datos de los alumnos (nombre, DNI, fotografía, ...)	<u>getObjectsByTitle()</u> : descubre el identificador del fichero alumnos.csv de la asignatura <u>getFileXML()</u> : descarga el fichero
Subir ficheros csv con los datos de asistencia	<u>addFile()</u> : añade un fichero a la plataforma. <u>updateFile()</u> : actualiza un fichero existente en la plataforma.

Tabla 3 - Servicios Web definitivos

Las comunicaciones de los servicios web se realizan a través del protocolo SOAP que, a grandes rasgos, es un protocolo basado en XML con el objetivo de realizar peticiones entre objetos, utilizando HTTP para el transporte de la información.

Para solicitar los datos hay que formar un sobre SOAP adaptado a la estructura que necesitan los distintos servicios y se recibe también otro XML que hay que procesar para recuperar la información.

En los servicios que vamos a utilizar, tenemos:

- **Login()**:
  - Parámetros de entrada
    - Client: valor fijo “docencia”.
    - Username: usuario de la plataforma de docencia virtual.
    - Password: clave del usuario indicado en Username.
  - Parámetros de salida
    - Sid: identificador de la sesión actual.

Petición	<pre> &lt;soapenv:Body&gt;   &lt;urn:login soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;client xsi:type="xsd:string"&gt;docencia&lt;/client&gt;   &lt;username xsi:type="xsd:string"&gt;XXXXXXXXX&lt;/username&gt;   &lt;password xsi:type="xsd:string"&gt;XXXXXXXXX&lt;/password&gt; &lt;/urn:login&gt; &lt;/soapenv:Body&gt; </pre>
Respuesta	<pre> &lt;SOAP-ENV:Body&gt;   &lt;ns1:loginResponse xmlns:ns1="urn:ilUserAdministration"&gt;   &lt;sid xsi:type="xsd:string"&gt;mbperisb65ihbk9orp1fm84b43::docencia&lt;/sid&gt; &lt;/ns1:loginResponse&gt; &lt;/SOAP-ENV:Body&gt; </pre>

Tabla 4 - Cuerpo mensaje SOAP Login. Solicitud y respuesta

- **getUserIdBySid():**
  - Parámetros de entrada
    - Sid: identificador de la sesión actual.
  - Parámetros de salida
    - Usr\_id: identificador del usuario que inició la sesión.

Petición	<pre> &lt;soapenv:Body&gt;   &lt;urn:getUserIdBySid soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;sid xsi:type="xsd:string"&gt;mbperisb65ihbk9orp1fm84b43::docencia&lt;/sid&gt; &lt;/urn:getUserIdBySid&gt; &lt;/soapenv:Body&gt; </pre>
Respuesta	<pre> &lt;SOAP-ENV:Body&gt;   &lt;ns1:getUserIdBySidResponse xmlns:ns1="urn:ilUserAdministration"&gt;   &lt;usr_id xsi:type="xsd:int"&gt;XXXXXX&lt;/usr_id&gt; &lt;/ns1:getUserIdBySidResponse&gt; &lt;/SOAP-ENV:Body&gt; </pre>

Tabla 5 - Cuerpo mensaje SOAP getUserIdBySid. Solicitud y respuesta

- **getObjectsByTitle():**
  - Parámetros de entrada
    - Sid: identificador de la sesión actual.

- Title: título del objeto que se busca. En este caso alumnos.csv.
- Usr\_id: identificador del usuario que inició la sesión.
- Parámetros de salida
  - Object xml: objeto xml con la descripción de todos los ficheros alumnos.csv a los que se tiene acceso.
    - Cada fichero tiene un elemento **References**, con un atributo **ref\_id**, que contiene la referencia que habrá que solicitar a Ilias para recuperar el fichero.
    - Se puede observar un atributo **parent\_id** por cada elemento References, con la referencia Ilias del directorio asistencia. Dicha referencia habrá que almacenarla para la posterior carga del fichero de sesiones.
    - Se puede observar también por cada fichero un elemento **element** con un atributo **type="crs"**. Este elemento identifica la asignatura a la que pertenece el fichero alumnos.csv. El atributo **ref\_id**, contiene, además, el identificador Ilias de la asignatura.

P	<pre> &lt;soapenv:Body&gt;   &lt;urn:getObjectsByTitle soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;sid xsi:type="xsd:string"&gt;mbperisb65ihbk9orp1fm84b43::docencia&lt;/sid&gt;   &lt;title xsi:type="xsd:string"&gt;alumnos.csv&lt;/title&gt;   &lt;user_id xsi:type="xsd:int"&gt;XXXXXX&lt;/user_id&gt; &lt;/urn:getObjectsByTitle&gt; &lt;/soapenv:Body&gt; </pre>
R	<pre> &lt;SOAP-ENV:Body&gt;   &lt;ns1:getObjectsByTitleResponse xmlns:ns1="urn:ilUserAdministration"&gt;     &lt;object_xml xsi:type="xsd:string"&gt;&lt;![CDATA[&lt;?xml version="1.0" encoding="utf-8"?&gt;&lt;!DOCTYPE Objects PUBLIC "-//ILIAS//DTD ILIAS Repositoryobjects//EN" "http://dv.ujaen.es/docencia/xml/ilias_object_4_0.dtd"&gt;&lt;!-- -Export of ILIAS objects--&gt; &lt;Objects&gt; &lt;Object type="file" obj_id="911313"&gt; &lt;Title&gt;alumnos.csv&lt;/Title&gt; &lt;Description/&gt; </pre>

```

<Owner>11505</Owner>
<CreateDate>2016-04-07 18:25:24</CreateDate>
<LastUpdate>2016-04-07 18:25:24</LastUpdate>
<ImportId/>
<Properties>
<Property name="fileSize">854</Property>
<Property name="fileExtension">csv</Property>
<Property name="fileVersion">1</Property>
</Properties>
<References ref_id="701931" parent_id="449295" accessInfo="granted">
<TimeTarget type="0">
<Timing starting_time="1460046325" ending_time="1460046325" visibility="0"/>
<Suggestion starting_time="1460046325" ending_time="1460046325"
changeable="0" earliest_start="1460046325" latest_end="1460066100"/>
</TimeTarget>
<Operation>visible</Operation>
<Operation>read</Operation>
<Path>
<Element ref_id="1" type="root">Espacios</Element>
<Element ref_id="30" type="cat">Escuela Politécnica Superior de
Linares</Element>
<Element ref_id="70" type="cat">Ingeniería de Telecomunicaciones</Element>
<Element ref_id="18693" type="cat">5º Curso</Element>
<Element ref_id="133664" type="crs">Aplicaciones para Telefonía
Móvil</Element>
<Element ref_id="449295" type="fold">asistencia</Element>
</Path></References></Object>
...
<Object type="file"
obj_id="801981"><Title>alumnos.csv</Title><Description/><Owner>11505</Own
er><CreateDate>2015-06-05 19:17:28</CreateDate><LastUpdate>2015-06-05
19:17:28</LastUpdate><ImportId/><Properties><Property
name="fileSize">854</Property><Property
name="fileExtension">csv</Property><Property
name="fileVersion">1</Property></Properties><References ref_id="616837"
parent_id="384129" accessInfo="granted"><TimeTarget type="0"><Timing

```

```

starting_time="1433524649" ending_time="1433524649"
visibility="0"/><Suggestion starting_time="1433524649"
ending_time="1433524649" changeable="0" earliest_start="1433524649"
latest_end="1433541300"/></TimeTarget><Operation>visible</Operation><Oper
ation>read</Operation><Path><Element ref_id="1"
type="root">Espacios</Element><Element ref_id="30" type="cat">Escuela
Politécnica Superior de Linares</Element><Element ref_id="77" type="cat">I.T.
de Telecomunicaciones: Esp. Telemática</Element><Element ref_id="569"
type="cat">3ª Curso</Element><Element ref_id="9904" type="crs">Transporte
de Datos</Element><Element ref_id="384129"
type="fold">asistencia</Element></Path></References></Object></Objects>]]<
/object_xml>
  </ns1:getObjectsByTitleResponse>
</SOAP-ENV:Body>

```

Tabla 6 - Cuerpo mensaje SOAP *getObjectsByTitle*. Solicitud y respuesta

- **getFileXML()**:
  - Parámetros de entrada
    - Sid: identificador de la sesión actual.
    - Ref\_id: referencia llias del fichero, recuperada con el servicio anterior.
    - Usr\_id: identificador del usuario que inició la sesión.
  - Parámetros de salida
    - filexml: objeto xml con el fichero codificado en formato bajo el tag Content, en formato **BASE64**<sup>1</sup>.

<sup>1</sup> BASE64: Sistema de codificación de binario a texto que utiliza un conjunto de 64 caracteres para representar 6 bits de datos.

Petición	<pre> &lt;soapenv:Body&gt;   &lt;urn:getFileXML soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;sid xsi:type="xsd:string"&gt;mbperisb65ihbk9orp1fm84b43::docencia&lt;/sid&gt;   &lt;ref_id xsi:type="xsd:int"&gt;701931&lt;/ref_id&gt;   &lt;attachment_mode xsi:type="xsd:int"&gt;XXXXXX&lt;/attachment_mode&gt; &lt;/urn:getFileXML&gt; &lt;/soapenv:Body&gt; </pre>
Respuesta	<pre> &lt;SOAP-ENV:Body&gt;   &lt;ns1:getFileXMLResponse xmlns:ns1="urn:ilUserAdministration"&gt;     &lt;filexml      xsi:type="xsd:string"&gt;&lt;![CDATA[&lt;?xml      version="1.0" encoding="utf-8"?&gt;&lt;!DOCTYPE      File      PUBLIC      "-//ILIAS//DTD FileAdministration//EN" "http://dv.ujaen.es/docencia/xml/ilias_file_3_8.dtd"&gt;&lt;!-- -Exercise Object--&gt;&lt;File obj_id="il_1946_file_911313" version="1" size="854" type="text/comma-separated- values"&gt;&lt;Filename&gt;alumnos.csv&lt;/Filename&gt;&lt;Title&gt;alumnos.csv&lt;/Title&gt;&lt;De scription/&gt;&lt;Rating&gt;0&lt;/Rating&gt;&lt;Content mode="PLAIN"&gt;IExpc3RhcyBkZSBjb...SI7IFPtDQo=&lt;/Content&gt;&lt;Versions&gt;&lt; Version          id="1"          date="1460046324" usr_id="il_1946_usr_11505"/&gt;&lt;/Versions&gt;&lt;/File&gt;]]&gt;&lt;/filexml&gt;   &lt;/ns1:getFileXMLResponse&gt; &lt;/SOAP-ENV:Body&gt; </pre>

Tabla 7 - Cuerpo mensaje SOAP getFileXML. Solicitud y respuesta

- **addFile():** se utilizará solamente la primera vez que se cargue a Ilias el fichero de AsistenciaPracticas.csv de cada asignatura para crear un fichero con únicamente columnas de cabecera (no se muestra la respuesta, para evitar cargar datos sin sentido a la plataforma):
  - Parámetros de entrada
    - Sid: identificador de la sesión actual.
    - Target id: directorio al que se va a subir el fichero de asistencia. Recuperado de la llamada **getObjectsByTitle()**.
    - Xml: fichero a subir, en formato **BASE64**.
  - Parámetros de salida
    - filexml: mismo fichero de entrada, con su referencia Ilias.

Petición	<pre> &lt;soapenv:Body&gt;   &lt;urn:addFile soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;sid xsi:type="xsd:string"?&lt;/sid&gt;   &lt;target_id xsi:type="xsd:int"?&lt;/target_id&gt;   &lt;xml xsi:type="xsd:string"?&lt;/xml&gt; &lt;/urn:addFile&gt; &lt;/soapenv:Body&gt; </pre>
----------	--

Tabla 8 - Cuerpo mensaje SOAP addFile. Solicitud

- **updateFile()** se utilizará cada vez que se quieran añadir datos de asistencia a un fichero existente. El servicio funciona por reemplazo del existente (no se muestra la respuesta, al no tener referencia al fichero):
  - o Parámetros de entrada
    - Sid: identificador de la sesión actual.
    - Ref\_id: Referencia al fichero de asistencia. Recuperado de la llamada **addFile()**.
    - Xml: fichero a subir, en formato **BASE64**.
  - o Parámetros de salida
    - filexml: mismo fichero de entrada.

Petición	<pre> &lt;soapenv:Body&gt;   &lt;urn:updateFile soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;sid xsi:type="xsd:string"?&lt;/sid&gt;   &lt;ref_id xsi:type="xsd:int"?&lt;/ref_id&gt;   &lt;xml xsi:type="xsd:string"?&lt;/xml&gt; &lt;/urn:updateFile&gt; &lt;/soapenv:Body&gt; </pre>
----------	--

Tabla 9 - Cuerpo mensaje SOAP updateFile. Solicitud

SOAP no está soportado de forma nativa en Android, por lo que hay que buscar una librería compatible, libre y a poder ser gratuita que implemente las funciones básicas de comunicaciones a través de SOAP o bien habría que diseñar dicha funcionalidad desde cero.

Haciendo una búsqueda en los principales repositorios de software libre, se encuentra dentro de GitHub, la librería kSoap2 (URL <http://simpligility.github.io/ksoap2-android/index.html>). Una versión anterior de dicha librería también es utilizada en el

proyecto indicado en *Antecedentes*, por lo que se decide que también se va a utilizar kSoap2 en la aplicación objeto de este proyecto.

### 1.2.2 Modelo de Datos

Es conveniente almacenar los datos de las asignaturas y los alumnos de manera local, para permitir operar en cualquier lugar sin perjuicio de la cobertura que de la que se disponga. Para ello, se decide almacenar la información de la plataforma de docencia virtual de manera local. Se plantean tres alternativas:

- Ficheros de texto plano.
- Ficheros XML.
- Base de datos.

La opción de ficheros de texto plano es descartada rápidamente debido a la dificultad añadida de manejar el texto, además de los datos.

Trabajar con ficheros XML es cómodo debido a las funciones de XML que proporciona la versión Android de Java, junto con la tecnología XPath para acceder a los nodos de manera directa, tal y como si se hiciera una petición a una base de datos. Esta solución tiene la contrapartida de que si en el diseño del almacenamiento conviene separar datos en distintas entidades, la gestión de la búsqueda es más compleja y recae en las fases de implementación y mantenimiento de la aplicación. Además, a medida que los ficheros van ganando en tamaño, la búsqueda se va tornando cada vez más lenta.

La solución de base de datos nos permite un acceso rápido a los datos que se desean encontrar, junto con una gestión desatendida de las distintas entidades de datos, que se almacenarían en tablas separadas.

Por otro lado, el usuario también generará información de manera local que tendrá que ser almacenada: grupos, sesiones y datos de asistencia a las sesiones.

En principio, toda esta información puede estar contenida en la propia base de datos, ya que está directamente relacionada con la que se descarga de la plataforma de docencia virtual. Sin embargo, si se analiza si es conveniente crear una tabla específica con los datos de asistencia a las sesiones, se observa que sería compleja de mantener ya que tendría los datos de todas las sesiones en una misma tabla, junto con las sesiones recuperadas, por lo que se opta por hacerlo de forma individualizada por sesión, generando un fichero csv por cada una, con un formato que permitirá ser adjuntado al final del fichero existente en Ilias de manera directa, ya que se gana en sencillez y se trabaja también con el acceso a ficheros de Android.

Debido a todas las ventajas que aporta a la hora de gestionar datos estáticos, se opta por almacenar la información local en una base de datos. Para los datos de las sesiones, se opta por utilizar ficheros en formato de texto plano separado por comas.

Por último, la aplicación va a soportar la identificación del alumno a través de NFC. Por un lado, el tutor dispondrá necesariamente de un terminal móvil, ejecutando la aplicación AsistenciaUJA, por lo que, en principio, podría utilizarse cualquiera de los modos de comunicación NFC:

- Modo activo: el alumno se identificaría con su propio dispositivo móvil. Este procedimiento tiene como ventaja que prácticamente siempre, el alumno va a tener el móvil, debido a los hábitos sociales; pero presenta dos desventajas importantes. Por un lado, habría que diseñar dos aplicaciones móviles para realizar la comunicación, con lo que el esfuerzo se duplica y, por otro, sería necesario que el alumno siempre tuviera instalada dicha aplicación para el correcto funcionamiento del sistema, lo cual no se puede presuponer.
- Modo pasivo: el alumno dispone de una tarjeta NFC con la que identificarse. Con este sistema, solamente se necesita una aplicación (la del tutor) y el alumno quedaría identificado acercando la tarjeta al terminal del mismo.

Debido a que solamente se necesita una aplicación para que funcione el sistema y a que no es necesario forzar a que el alumno instale una aplicación adicional, se decide utilizar NFC en modo pasivo.

#### *1.2.2.1 Base de Datos*

Según lo expuesto en el apartado anterior, para que la aplicación funcione correctamente, se observaran las siguientes entidades y relaciones entre ellas:

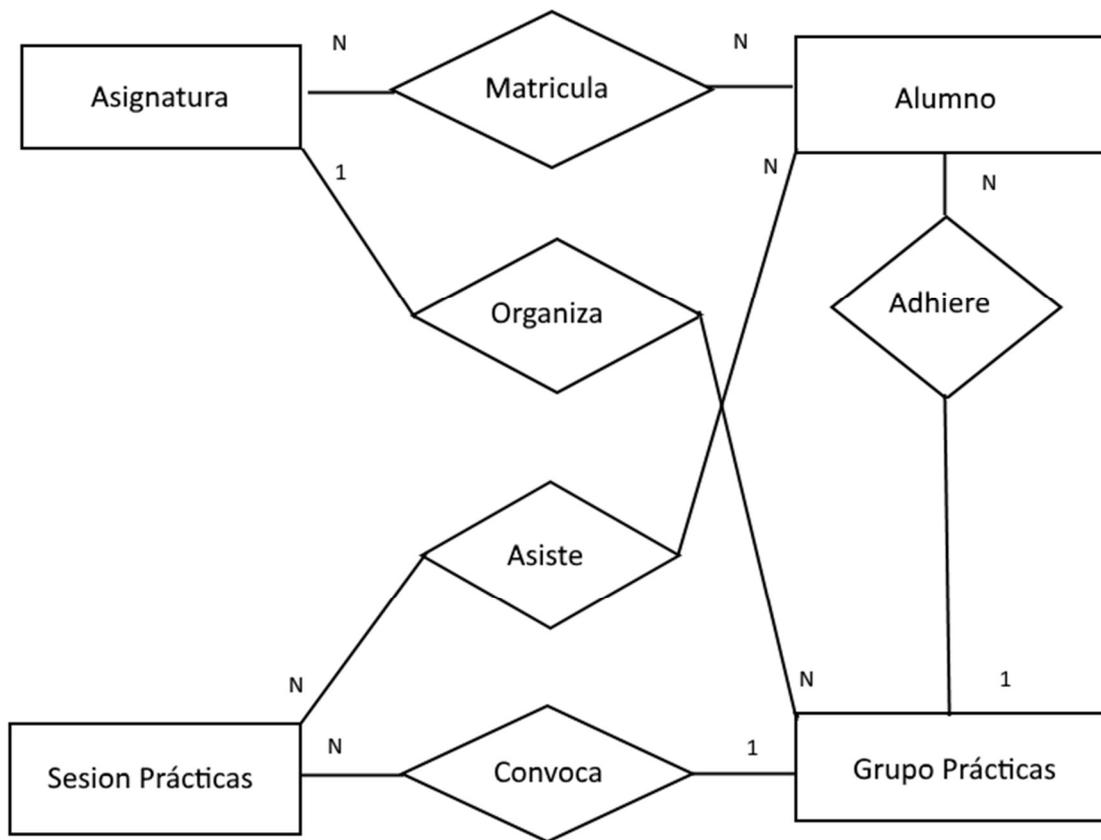


Figura 6 - Modelo Entidad - Relación

Para soportar dicho modelo y la funcionalidad que se desea, la base de datos debe almacenar:

- Asignaturas
  - o IdAsignatura: referencia interna de la asignatura.
  - o Nombre: nombre de la asignatura.
  - o refAsistencia: referencia llias al directorio Asistencia, de donde se descargará el fichero de alumnos y en el que se almacenarán los datos de asistencia de la misma.
  - o rutaFicheros: ruta local donde se almacenarán los ficheros de las sesiones.
- Alumnos
  - o IdAlumno: referencia interna del alumno.
  - o Dni: dni del alumno.
  - o Nombre: nombre del alumno.
  - o Ape1: primer apellido del alumno.
  - o Ape2: segundo apellido del alumno.
- Grupos de Practicas
  - o idGrupo: identificador interno del grupo.

- Nombre: nombre que se le da al grupo para facilitar su identificación.
  - Descripción: descripción del grupo.
  - Editable: valor lógico para indicar si el grupo puede ser editado/borrado. Si se han confirmado datos de asistencia de un determinado grupo en Ilias, no tiene sentido que el grupo se pueda borrar posteriormente.
  - Borrado: valor lógico para indicar si el grupo ha sido borrado, por si se implementa una posible recuperación de datos borrados.
  - idAsignatura: identificador de la asignatura a la que pertenece el grupo.
  - Día: día de la semana en el que el grupo tiene sus sesiones de prácticas.
  - Hora: hora en la que el grupo tiene sus sesiones de prácticas.
- Sesiones de Prácticas
- idSesion: identificador interno de la sesión, por grupo.
  - idGrupo: identificador del grupo al que pertenece la sesión.
  - FechaCreación: fecha en la que se creó la sesión de prácticas, para realizar un seguimiento.
  - FechaSesión: fecha para la que está programada la sesión.
  - HoraSesión: hora a la que está programada la sesión.
  - RutaFicheroLocalSesión: ruta completa al fichero csv almacenado localmente de la sesión, para poder ver su contenido y subirlo a Ilias.
  - idIliasFicheroSesion: ruta del fichero de la sesión en Ilias, en previsión de necesitar un posible registro de cada una.
  - Estado: estado de la sesión. Solamente se permitirán los siguientes valores:
    - CREADA: la sesión acaba de ser creada.
    - ABIERTA: la sesión ha sido abierta por el tutor.
    - CERRADA: la sesión ha sido cerrada por el tutor.
    - SUBIDA: la sesión ha sido subida a Ilias.
    - INDEFINIDA: valor por defecto del campo. No aporta información
    - En la práctica solamente se utilizarán los valores de CREADA y SUBIDA, ya que el resto únicamente suponen control adicional sobre las acciones del tutor; pero no aportan valor real al control de asistencia. Se mantiene en el diseño para futuras ampliaciones.

- Duración: duración en minutos de la sesión de prácticas.
- Relación entre grupos de prácticas y alumnos
  - idAlumno: referencia interna del alumno.
  - idAsignatura: referencia interna de la asignatura. Se utilizará para conocer si un determinado alumno está matriculado en la asignatura, y mostrarlos así correctamente en grupos de prácticas y sesiones.
  - idGrupo: referencia interna del grupo. Se utilizará para saber el grupo al que se ha adherido un determinado alumno.
- Referencias llias a los recursos que genera la aplicación en línea.
  - Nombre: nombre del recurso en llias.
  - IdAsignatura: referencia a la asignatura a la que está asociado el recurso.
  - idllias: referencia en llias a dicho recurso.

Por tanto, el modelado de las tablas queda así:

- Asignaturas (idAsignatura, nombre, refAsistencia, rutaFicheros)
- Alumnos (idAlumno, dni, nombre, ape1, ape2)
- GrupoPracticas (idGrupo, nombre, descripción, editable, borrado, idAsignatura, día, hora)
- SesionPracticas (idSesion, idGrupo, fechaCreacion, fechaSesion, horaSesion, rutaLocalFicheroSesion, idlliasFicheroSesion, estado, duracion)
- AlumnoAsignaturaGrupo (idAlumno, idAsignatura, idGrupo)
- llias (nombre, idAsignatura, idllias)
- SesionPracticas – idGrupo → GrupoPracticas
- GrupoPracticas – idAsignatura → Asignaturas
- AlumnoAsignaturaGrupo – idAlumno → Alumnos
- AlumnoAsignaturaGrupo – idAsignatura → Asignaturas
- AlumnoAsignaturaGrupo – idGrupo → GrupoPracticas
- llias – idAsignatura → Asignaturas.

### 1.2.2.2 Fichero de sesión

Se decidió durante el análisis de Servicios Web que los datos de asistencia se subirán a docencia virtual en un único fichero en formato de texto separado por comas que recoja las entradas y salidas de los alumnos a las sesiones de prácticas y que, además, permita discernir si la asistencia del alumno a una determinada sesión es debido a que es de su propio grupo o a si está recuperando asistencia.

El formato más apropiado para el fichero será: un registro por cada alumno perteneciente al grupo de la sesión, más un registro por cada alumno adicional que se

haya presentado a la misma, aunque no pertenezca al grupo, con el fin de contabilizar recuperaciones de asistencia. Dicho registro presentará el siguiente contenido:

- GrupoSesion: Identificador del grupo para el que fue creada la sesión actual.
- Sesión: Identificador de la sesión.
- Fecha: Fecha en la que se celebra la sesión.
- Hora: Hora a la que se celebra la sesión.
- GrupoAlumno: Grupo al que pertenece el alumno. Si dicho grupo es diferente al de la sesión, significa que dicho alumno está recuperando asistencia.
- IdAlumno: Identificador Ilias del alumno. Útil para tareas de mantenimiento de la aplicación.
- DniAlumno: DNI del alumno. Es el único dato del alumno que se sube a docencia virtual, para preservar al máximo su privacidad. Realizando un filtrado por este campo, el tutor podrá comprobar el total de asistencias de un determinado alumno.
- Asistencia: valor lógico a true o false. Si es true, significa que el alumno ha asistido a la sesión y, por tanto, deberán aparecer datos en los campos HoraEntrada y HoraSalida. Es útil para tareas de mantenimiento y, si se realiza un filtrado por este campo y DniAlumno, se puede obtener el total de prácticas a las que el alumno ha asistido y contabilizar las veces a las que no se ha presentado en su propio grupo.
- HoraEntrada: Hora a la que el alumno entró a la sesión.
- HoraSalida: Hora a la que el alumno abandonó la sesión.

A la hora de contabilizar el total de asistencia a prácticas, se puede utilizar cualquier aplicación de hoja de cálculo para comprobar el total de asistencias de un determinado alumno, filtrando por su DNI. Además, se puede comprobar el total de sesiones recuperadas si se comparan los campos GrupoSesión y GrupoAlumno, entre otras operaciones.

El fichero resultante se vería así en un editor de texto común:

```
GrupoSesion;Sesion;Fecha;Hora;GrupoAlumno;IdAlumno;DniAlumno;Asistencia;HoraEntrada;HoraSalida
0;1;20160404;11:00;0;8;44039224C;false;;
0;1;20160404;11:00;-1;7;75110729C>true;12:10;12:10
0;1;20160404;11:00;-1;9;77336245X>true;12:09;12:09
0;1;20160404;11:00;0;5;75108888L>true;12:09;12:09
```

Figura 7 - Ejemplo de Fichero de sesión

Este tipo de fichero cumple con el objetivo de *guardado en un formato fácilmente editable*, establecido en el apartado 1.1.2.

### 1.2.2.3 Tarjeta NFC

La tarjeta NFC es necesaria para que el alumno se pueda identificar a sí mismo en cada sesión de prácticas, acercando la misma al terminal Android. Hay que definir los datos y su formato para que sean útiles y legibles por la aplicación.

Dicha tarjeta se puede escribir con cualquier terminal Android, es decir, no es necesario ningún tipo de hardware NFC especial, por lo que, se puede descargar al tutor de la obligación de escribir dichos datos y permitir que los mismos alumnos puedan realizar esta tarea. Para permitirles esto, la información a cargar debe ser algo conocido por los alumnos, tanto los que están familiarizados con la plataforma virtual de la Universidad de Jaén, como para los de nuevo ingreso. Atendiendo a esto, se ha optado por cargar las tarjetas NFC de la forma más sencilla posible con una única línea de texto, en la que aparezca en primer lugar el DNI del alumno, en el mismo formato en el que se está en el fichero alumnos.csv, separado por comas de cualquier otro dato que se quiera añadir, de tal forma que cuando se lea la tarjeta, se buscara al alumno en la base de datos por su DNI. Se recomienda añadir nombre y apellidos completos para poder identificar al dueño de la tarjeta en caso de pérdida.

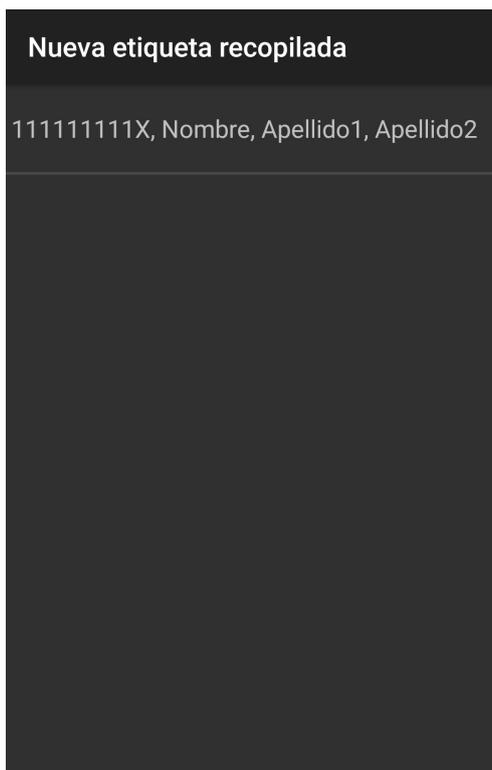


Figura 8 - Formato Tarjeta NFC

### 1.2.3 Aplicación móvil

Una vez que se ha definido la información que hay que almacenar y se le ha dado el formato más óptimo, el siguiente paso es diseñar y desarrollar la aplicación móvil que ejecutará el profesor para controlar la asistencia. En concreto se ha denominado a dicha aplicación **AsistenciaUJA**.

El pilar fundamental de las aplicaciones para Android son las actividades. Una aplicación Android se compone, habitualmente, de una o varias actividades que implementan individualmente una funcionalidad concreta y específica. Cada una de estas actividades suele tener asociado un fichero de diseño (aunque no es obligatorio) con los controles visuales necesarios para interactuar con el usuario; y se irán presentando al mismo en una secuencia controlada por el desarrollador para recabar los datos necesarios y operar acorde al objetivo de la actividad.

Las actividades pueden intercambiar información entre ellas. Para ello se pueden seguir dos patrones:

- Estándar: la actividad padre incluye los datos en la llamada a la actividad hija mediante los "extras", que son una lista de parámetros organizados por nombre y valor que la actividad hija puede recuperar de forma nativa. Los valores intercambiados solamente pueden ser tipos básicos y cadenas de texto: si se desea intercambiar objetos, hay que serializarlos primero. La principal ventaja de este sistema es que está directamente soportado en la API Android y es fácil de utilizar; pero en contrapartida tiene dos desventajas muy significativas: como se ha dicho, hay que serializar y deserializar los objetos para intercambiarlos, con lo que la dificultad y el tiempo de desarrollo se incrementan y, además, no hay forma de recuperar la información modificada en la clase hija, por lo que si se generan grupos de prácticas y se guardan en la base de datos, habría que hacer otra consulta en base de datos para cargar toda la información de nuevo.
- Singleton: este patrón se puede utilizar tanto en Android como en otros paradigmas de programación y consiste en suponer que, si solamente puede existir una instancia de los datos, y dichos datos se van a intercambiar y modificar entre actividades, tanto en las actividades padre como en las hijas; se puede generar una clase estática con dichos objetos entre sus miembros para que todas las actividades tengan acceso a ellos. Tiene como principal ventaja que se aceleran los tiempos de desarrollo y se facilita el mantenimiento ya que no hay que preocuparse de la serialización de los objetos y como contrapartida, que no es el método recomendado para el intercambio de información y que el desarrollador se tiene que preocupar de gestionar la integridad de los datos.

Debido a que en la aplicación solamente puede existir una instancia de la base de datos, solamente se puede trabajar a la vez con una asignatura, grupo de prácticas y/o sesión seleccionados y la mayor facilidad a la hora de intercambiar información al obviar la serialización y deserialización de objetos provenientes de la base de datos, se

ha optado por utilizar el patrón Singleton para los intercambios de información entre actividades.

Para trabajar con bases de datos, Android incorpora de forma nativa el motor SQLite. SQLite es un ligero motor de código abierto, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla y cuyas principales ventajas son:

- No requiere el soporte de un servidor: SQLite no ejecuta un proceso para administrar la información, sino que implementa un conjunto de librerías encargadas de la gestión.
- No necesita configuración: Libera al programador de todo tipo de configuraciones de puertos, tamaños, ubicaciones, etc.
- Usa un archivo para el esquema: Crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad, ya que los datos de las aplicaciones Android no pueden ser accedidos por contextos externos.
- Es de Código Abierto: Es de dominio público, y está abierta a los desarrolladores, al igual que sus archivos de compilación e instrucciones de escalabilidad.

El funcionamiento nativo es muy sencillo, ya que Android incorpora en su API la clase SQLiteOpenHelper, que permite realizar transacciones con la base de datos obviando la complejidad adicional de confeccionar peticiones SQL. La aplicación no tiene necesidades específicas de una base de datos concreta, por lo que se utilizará el sistema nativo SQLite.

#### *1.2.3.1 Diagrama de clases*

Android está basado en Java, por lo que también sigue un paradigma de programación basado en objetos, y los utiliza para almacenar y mantener en memoria la información necesaria de las actividades y la base de datos. Tal y como se ha diseñado dicha base de datos, se puede crear una clase por cada una de las tablas definidas. Los registros del fichero de texto en el que se almacenarán los datos de asistencia de los alumnos también se pueden modelar como un objeto y, dicho fichero a su vez se puede concebir como una lista de dichos registros. De acuerdo a esto, el diagrama de clases sería:

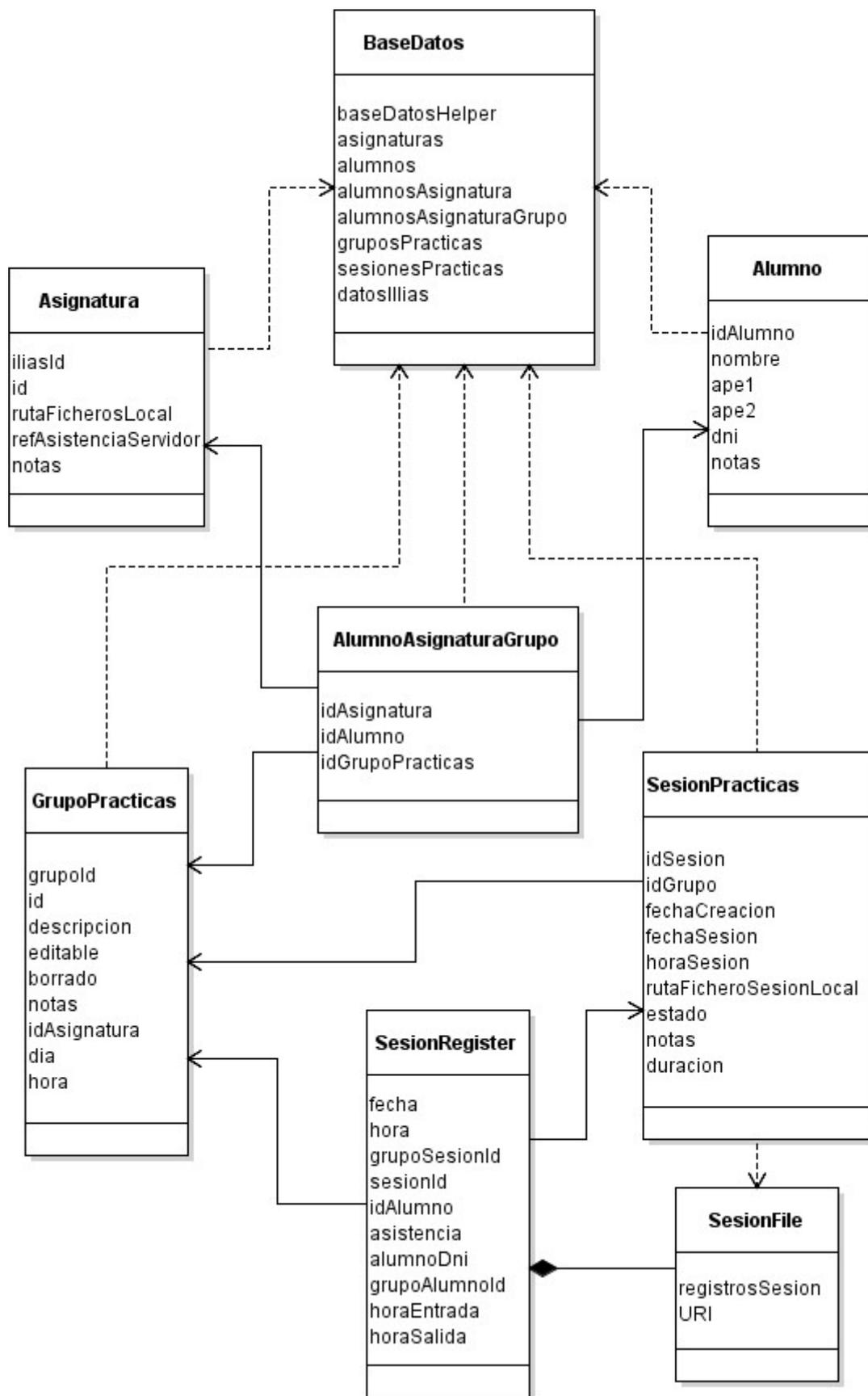


Figura 9 - Diagrama de clases

### 1.2.3.2 Diagrama de flujo

Una vez definido el formato de los datos en la aplicación, conviene visualizar el flujo en el que se debería ejecutar la aplicación:

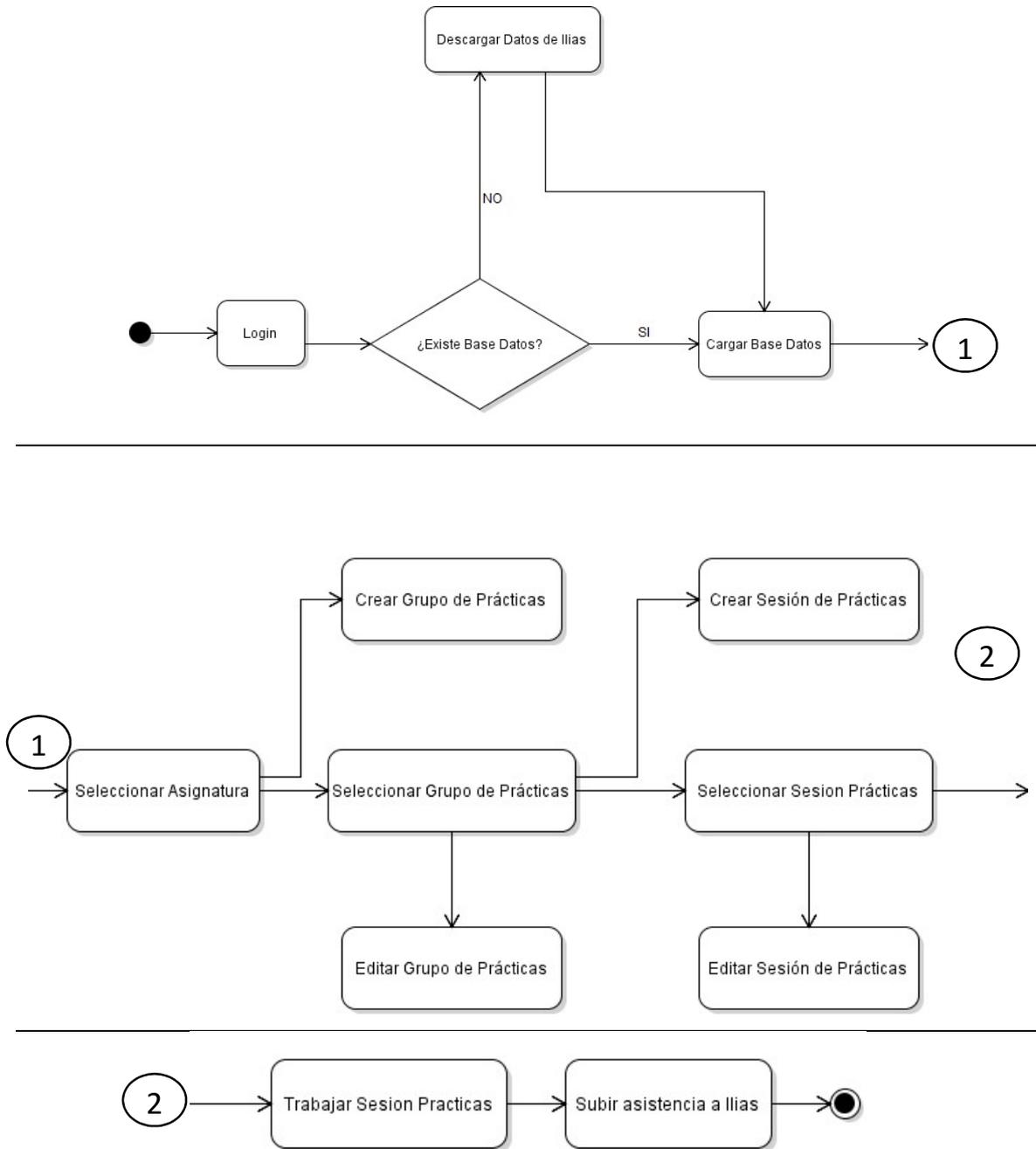


Figura 10 - Diagrama de Actividad

### 1.2.3.3 Login / Creación Base Datos

Para empezar, se debe decidir si la aplicación va a dar soporte a uno o a varios usuarios en cada terminal. Teniendo en cuenta que el uso de un dispositivo móvil es personal a día de hoy y que, para compartir el dispositivo, habría que compartir clave de

desbloqueo, no parece que se vaya a dar la posibilidad de que varios tutores operen con el mismo teléfono, por lo que se decide implementar una aplicación mono-usuario.

Inicialmente, la aplicación debe descubrir las asignaturas asociadas a un tutor, por lo que es necesario realizar al principio una operación de login, en la que se le envíen a Ilias las credenciales del usuario en cuestión y se le soliciten todos los ficheros alumnos.csv a los que el usuario tenga acceso, tal y como se solicita en el apartado *Objetivos – Toma de datos offline-actualización online*.

No se puede realizar una operación con servicios web en el mismo hilo de ejecución de una actividad, ya que se necesita una respuesta síncrona a cualquier petición realizada; pero debido al retardo en el tiempo de respuesta del servicio, Android interpreta que la aplicación ha dejado de responder y le da la opción al usuario de terminarla. Este comportamiento no es deseado, puesto que induce a confusión, por lo que se añade la dificultad de tener que implementar un objeto que permita realizar tareas asíncronas y esperar su respuesta y que evite que Android proponga terminar la aplicación. Para ello, se hará uso de una clase que herede de AsyncTask, la cual permite realizar operaciones de manera asíncrona; pero, además, se puede obligar al flujo de la actividad a que espere a que dicha operación asíncrona termine de ejecutarse y sin provocar que Android detecte un bloqueo del sistema.

Volviendo al hilo principal de la actividad, con la solicitud de ingreso se obtienen todas las asignaturas de las que un usuario es tutor ya que se necesitan dichos privilegios para acceder al fichero alumnos.csv. Además, con dicha petición, se obtiene el identificador del fichero que contiene los datos de los alumnos para su descarga, procesamiento e incorporación a la base de datos. Se hará todo en la misma actividad, ya que es a partir de aquí cuando se necesitará tener la base de datos cargada. Este sería el diagrama de caso de uso de esta actividad:

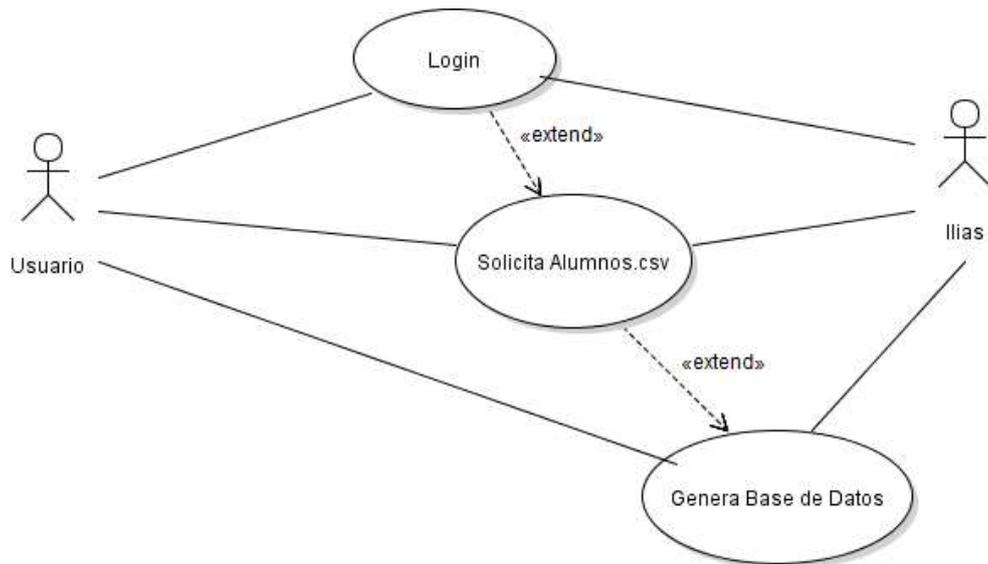


Figura 11 – Diagrama de Casos de uso Login

Para implementar dicha actividad, teniendo en cuenta los servicios Ilias a los que hay que acceder (orden de ejecución y parámetros), descritos en el apartado Servicios Web, se debe seguir el siguiente diagrama de flujo:

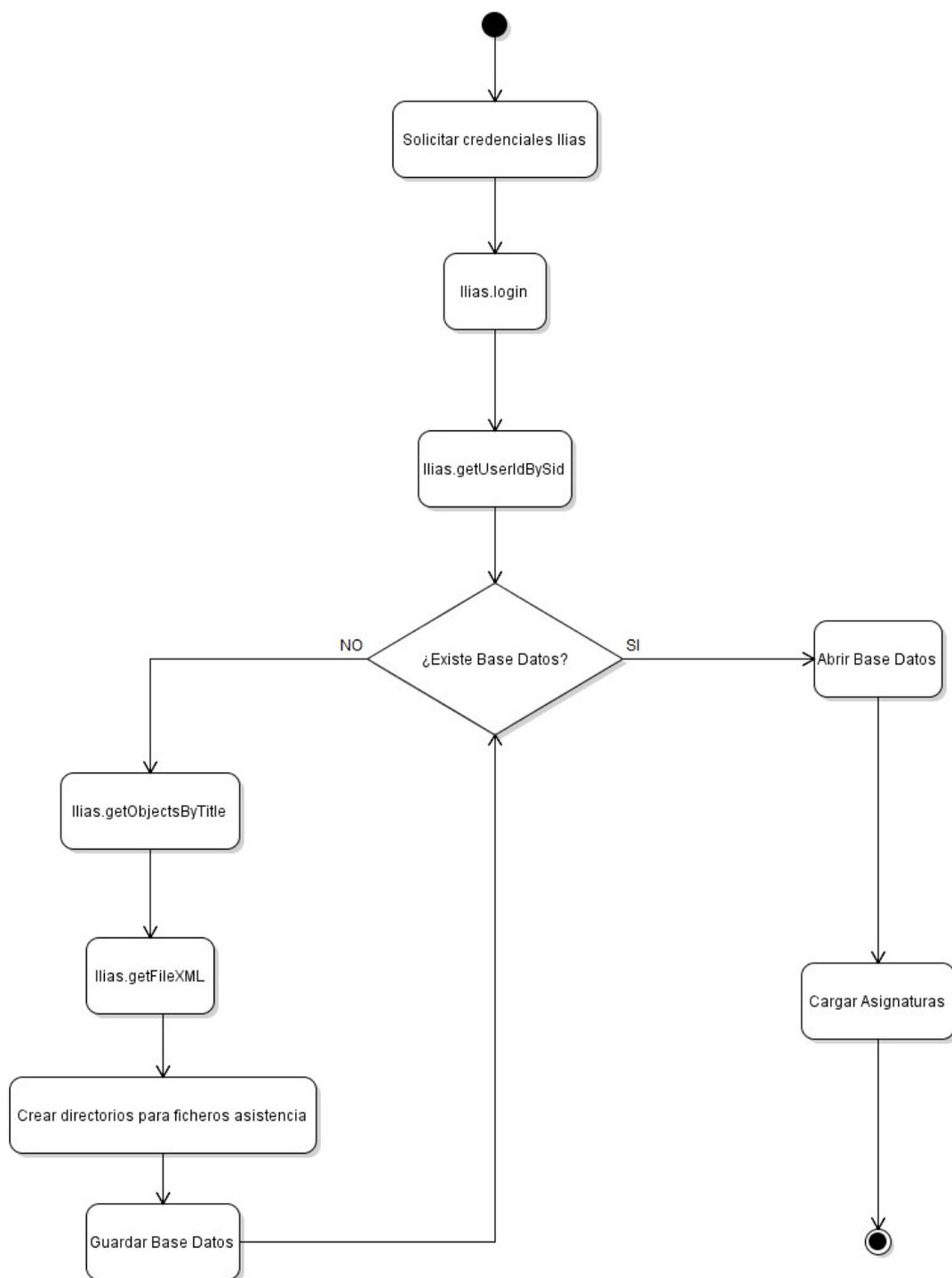
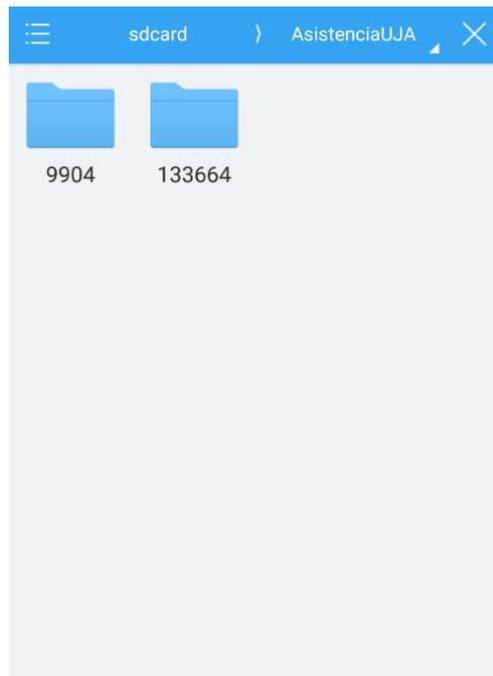


Figura 12 - Diagrama Flujo Login / Creación Base Datos

Por cada asignatura descubierta para el tutor en docencia virtual, se creará un directorio en el almacenamiento local cuyo nombre será el identificador de la asignatura en Ilias, dentro de la carpeta AsistenciaUJA:



*Figura 13 - Directorios de Asignaturas*

A partir de aquí, ya no será necesario volver a acceder a Ilias hasta que se deseen cargar los datos de asistencia de alguna sesión de prácticas, por lo que se puede trabajar en modo local una vez que exista la base de datos en el dispositivo. Sin embargo, se plantea el problema de la **privacidad**, ya que se mostrará información sensible de los alumnos, tal como su nombre completo, por lo que se hace necesario proteger el acceso a la aplicación por parte de usuarios no autorizados. Debido a ello, se decide ejecutar la actividad de login tanto si la base de datos ha de ser descargada de Ilias, como si ya se dispone de ella en local, cumpliendo así con el objetivo de *seguridad*.

Esta será la actividad de Login vista desde el dispositivo:

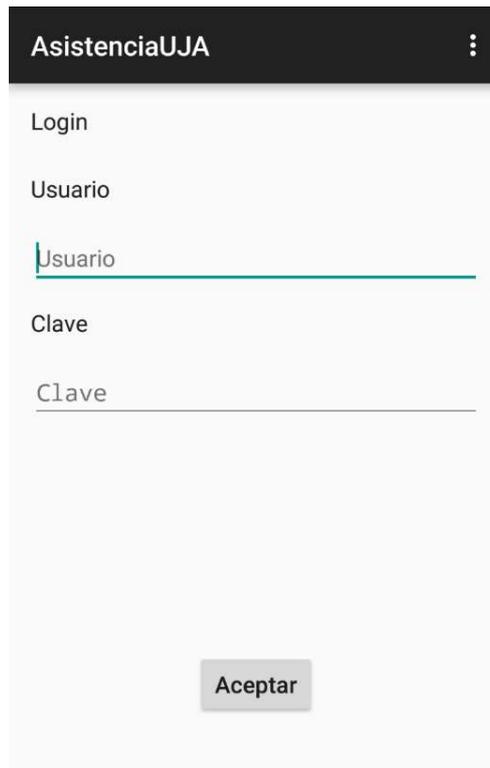


Figura 14 - Actividad Login

#### 1.2.3.4 Seleccionar Asignatura / Grupo Practicas

La siguiente actividad del diagrama Figura 10 - Diagrama de Actividad, indica que hay que proceder a la selección de una asignatura para operar con ella.

Desarrollar una actividad Android en la que únicamente se seleccione una asignatura para, acto seguido, ir a la siguiente actividad (que sería selección de grupo), no es rentable, ni desde el punto de vista del desarrollo, ni desde el punto de vista del usuario, ya que los datos de la asignatura son estáticos (vienen de Ilias y no tiene sentido modificarlos), supone añadir un paso extra que complica la gestión de actividades de la aplicación y reduce la usabilidad de cara al usuario. Por tanto, se van a agrupar bajo un mismo flujo, tanto la selección de asignatura, como la gestión de grupos.

El usuario va a poder seleccionar una asignatura de entre todas las que tiene en base de datos e, inmediatamente tras esa acción, aparecerán los grupos de prácticas que la forman. Por cada grupo de prácticas se mostrará en pantalla su identificador, el día y la hora de la semana en la que se han estipulado sus sesiones y el total de alumnos que pertenecen al mismo, para proporcionar la máxima información al usuario, mejorando así la experiencia.

Si el usuario desea cambiar de asignatura, va a poder ejecutar dicha acción desde la misma actividad Android de manera intuitiva y la aplicación cargará los grupos relacionados con la nueva selección.

Con los grupos de prácticas en pantalla, el usuario debe poder acceder de manera intuitiva a las siguientes acciones de gestión:

- **Editar Grupo:** para modificar los valores asignados al grupo durante su creación.
- **Ver Alumnos:** para ver los alumnos pertenecientes al grupo de prácticas en cuestión y darlos de alta o de baja en el grupo.
- **Ver Sesiones:** para ver las sesiones de prácticas previstas para el grupo.

Si la asignatura no tuviera ningún grupo de prácticas aún, se va a poder lanzar la acción de **crear un nuevo grupo de prácticas** desde la misma actividad.

Atendiendo a esto, el diagrama de casos de uso de esta actividad agrupada sería:

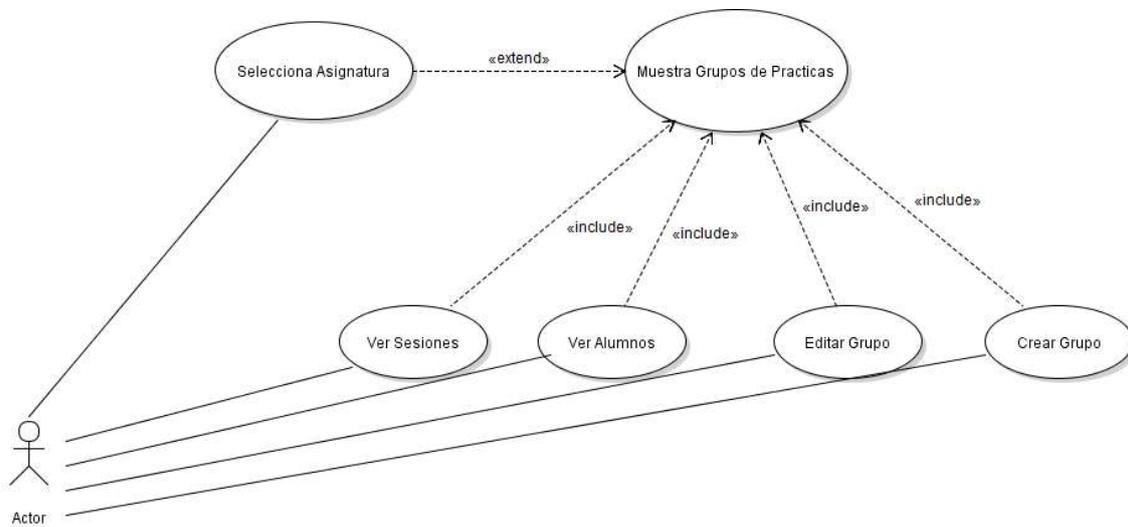


Figura 15 – Diagrama de Casos de Uso Seleccionar Asignatura / Grupo Prácticas

Y su diagrama de flujo sería el siguiente:

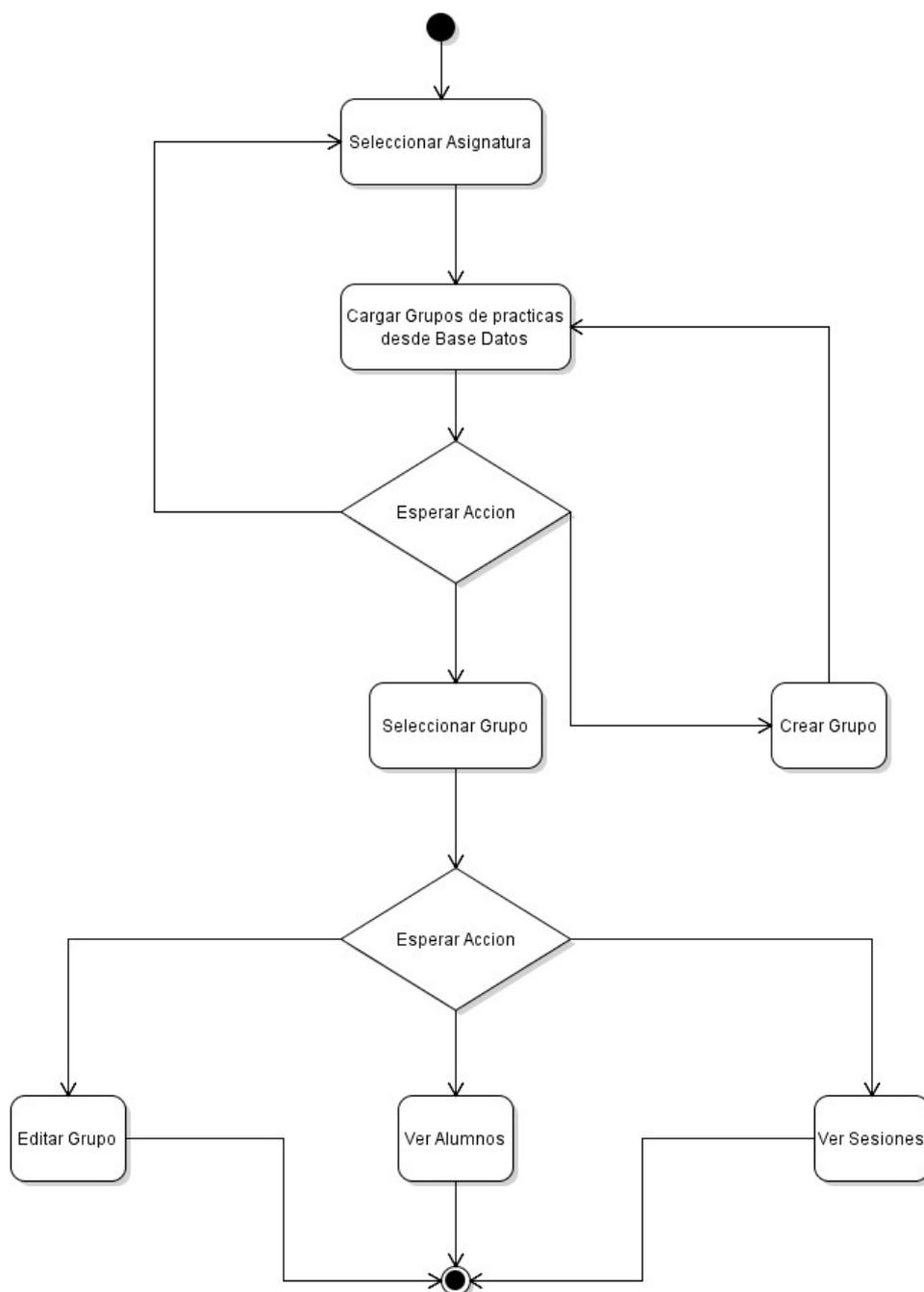


Figura 16 - Diagrama Flujo Seleccionar Asignatura / Grupo Prácticas

### 1.2.3.5 Seleccionar Asignatura / Grupo Practicas

Adicionalmente, es conveniente incluir una opción global para resetear todo el sistema, con miras a cuando se cambia de año lectivo, ya que habrá que generar nuevos grupos, volver a descargar los alumnos. Hay que tener en cuenta que esta acción es crítica, ya que realiza un borrado total de los datos de la aplicación, por lo que debe estar protegida de accesos no autorizados y por ello, solamente se puede realizar una vez que el usuario ha conseguido pasar la actividad login con éxito. Esta acción se va a poder realizar desde esta misma actividad y borrará todos los registros de la base de datos: asignaturas, grupos de prácticas, sus alumnos y sesiones asociadas, junto con

los ficheros en texto plano de la sesión y sus directorios adjuntos. Es recomendable no ocupar espacio de pantalla con esta acción, ya que se utilizará una vez al año y el espacio de visionado en un móvil es limitado.

La actividad Android que implementará todas estas acciones estará basada en un control NavigationDrawer, que implementa y gestiona un panel de navegación que se oculta automáticamente. En él se mostrarán todas las asignaturas y, en función de la que se seleccione, se cargarán sus grupos de prácticas en la ventana principal de la actividad.

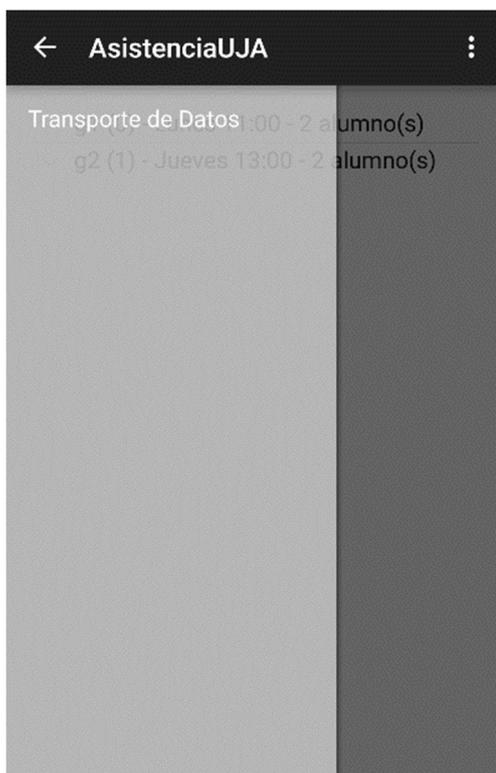


Figura 17 - Panel de Navegación Asignaturas



Figura 18 – Asignatura con grupos

Los grupos se mostrarán en una lista expandible, de tal forma que cuando se seleccione uno cualquiera, se desplegarán las opciones de edición de grupo, de alumnos y revisión de sesiones.

Como se puede observar en la actividad de la

Figura 18 – Asignatura con grupos, en la barra de acción hay una opción a la vista denominada “Crear Grupo”. Si el usuario pulsa sobre ella, se lanzará la actividad a describir en el punto Crear Grupo de Prácticas.

La acción de resetear base de datos se mostrará como una opción dentro de la susodicha barra de acción, de forma que no ocupe espacio de visualización efectivo.

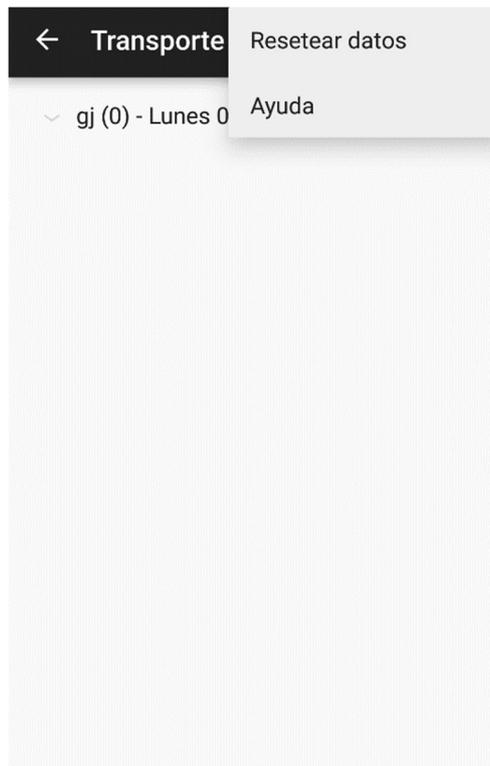


Figura 19 - Reseteo de Datos en Barra de Acción

#### 1.2.3.6 Crear Grupo de Prácticas

En el estado inicial, el sistema no dispone de ningún grupo de prácticas para las asignaturas, ya que su creación es local y no depende de Ilias. Por tanto, la aplicación debe proveer al usuario de una actividad capaz de crearlos.

Los datos a introducir por cada grupo de prácticas serán los indicados en el apartado Base de Datos, salvo los valores lógicos que implican estados del grupo y el identificador interno, que se genera automáticamente. Una vez que el usuario confirme la información del grupo, habrá que comprobar que no haya grupos con identificador sin indicar y que la información de fecha y hora esté introducida.

El diagrama de flujo será el siguiente:



*Figura 20 - Diagrama de Flujo de Crear Grupo*

La actividad que implemente esta tarea será un formulario simple que solicite los datos:

← Crear Grupo Practicas

Crear Grupo

Identificador de Grupo

Ejemplo

Descripcion del Grupo

Ejemplo

Dia de la semana Hora

Lunes

Aceptar

*Figura 21 - Actividad Crear Grupo*

Cuando el usuario pulsa sobre el botón aceptar, la actividad comprueba que la información esté correctamente introducida. Si es así, inserta la información en la base de datos y vuelve a la actividad de Seleccionar Asignatura / Grupo Practicas. En caso contrario, muestra una ventana emergente con el error para que el usuario rectifique la información.

Una vez introducida la información correctamente, la aplicación generará un directorio con el identificador interno del grupo, dentro de la carpeta de la asignatura para la que fue creado:

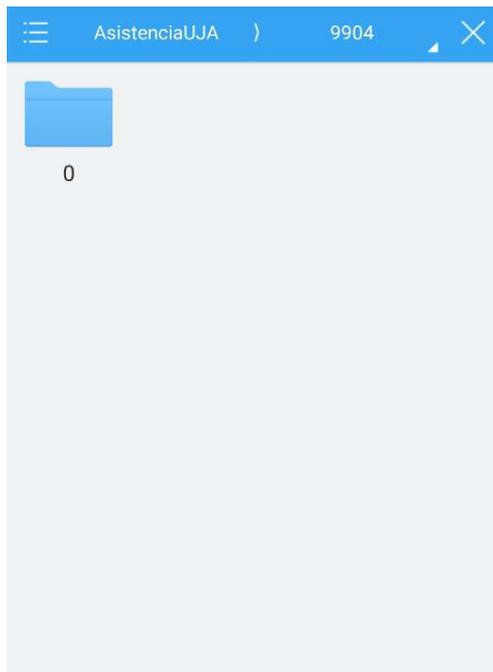


Figura 22 - Directorio Grupo Practicas

### 1.2.3.7 Editar Grupo de Prácticas

Cuando la asignatura ya dispone de grupos (véase Figura 18 – Asignatura con grupos), la aplicación permite dos tipos de operaciones de edición sobre ellos: editar la información actual del grupo y editar los alumnos que forman el grupo.

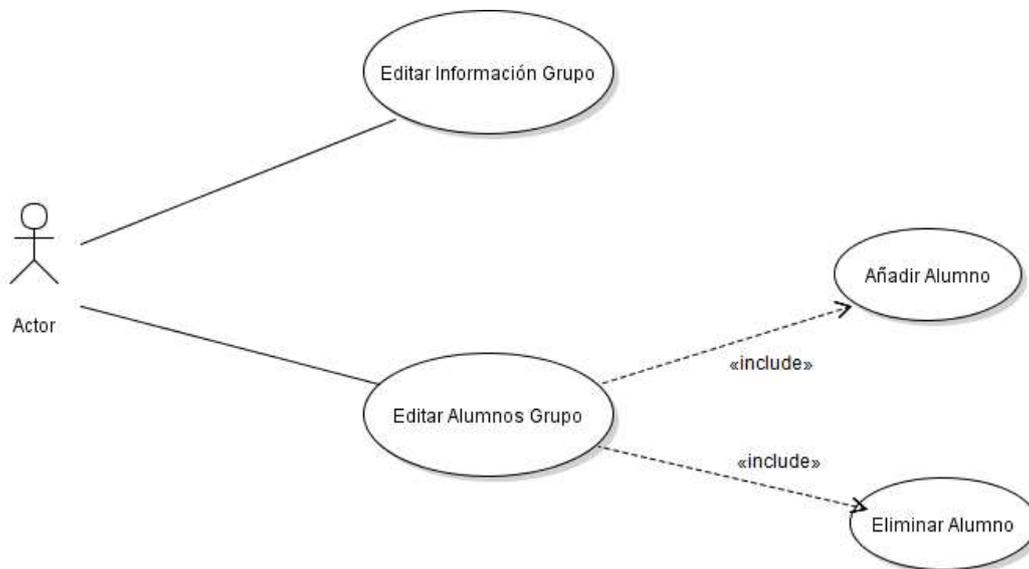


Figura 23 - Diagrama de Casos de Uso Editar Grupo

Cualquiera de estas dos acciones se puede seleccionar en la lista desplegable que aparece en la actividad de la

Figura 18 – Asignatura con grupos, tras pulsar sobre un grupo de prácticas concreto.

Si se selecciona la opción **“Editar Grupo”** se mostrará la misma actividad que en el apartado Crear Grupo de Prácticas, pero con la información del grupo en los campos correspondientes, lista para ser modificada. Todo el funcionamiento de la actividad será el mismo que en el anterior apartado.

Si se selecciona la opción **“Ver Alumnos”**, la aplicación mostrará una actividad en la que aparecerán todos los alumnos de la asignatura. Por cada alumno se mostrará:

- Nombre completo.
- Grupo al que pertenece.
- Casilla de verificación para añadir o eliminar al alumno del grupo.

Por cada vez que el tutor añade o elimina a un alumno del grupo, activando o desactivando respectivamente su casilla de verificación, la operación quedará registrada en la base de datos. Se ha diseñado de esta forma debido al funcionamiento de Android para gestionar las interrupciones tipo llamadas, cambios de aplicación, etc, ya que directamente se destruye la actividad actual y al volver, se crea de nuevo desde cero. Si no se guardara la información en cada interacción y se produjera una llamada, las acciones se perderían.

El diagrama de flujo de la actividad, por tanto, será:

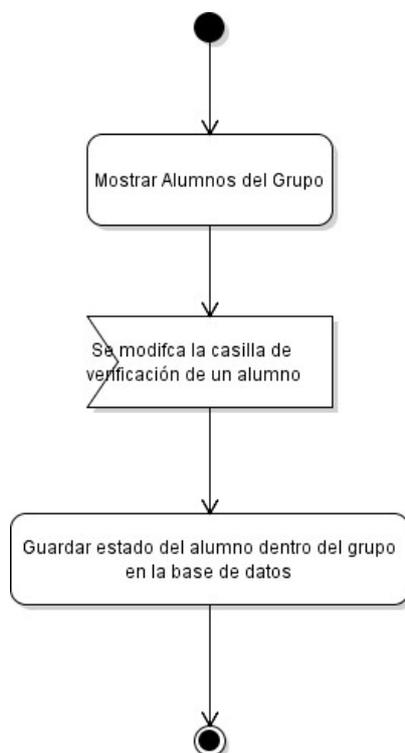


Figura 24 - Diagrama de Flujo Ver Alumnos

Para mejorar la experiencia del usuario, se ha añadido la funcionalidad de que, si un alumno en concreto ya pertenece a otro grupo, éste aparece resaltado en rojo y su casilla de verificación está desactivada, de forma que, si se desea incluirlo en el grupo

actual, hay que darlo de baja primero en su grupo de origen, realizando así de manera implícita una confirmación del cambio.

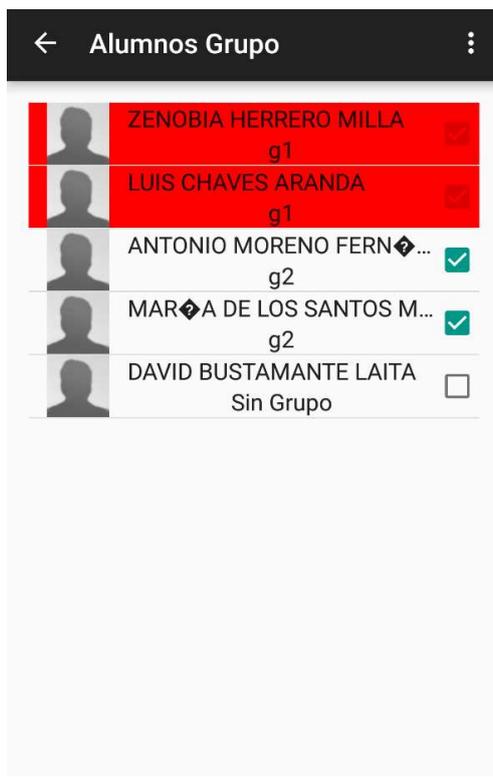


Figura 25 - Actividad Ver Alumnos

Se puede observar que en el diseño de la actividad se muestra una foto anónima por cada alumno de la asignatura. Al principio se diseñó así por si se solucionaba el problema de los permisos de Ilias y se podía acabar accediendo a las fotografías de los alumnos a través de los servicios web. Como no ha sido posible, se ha optado por mantener esta característica, para minimizar el impacto en el desarrollo de una futura ampliación que permita esta característica.

#### 1.2.3.8 Seleccionar Sesión de Prácticas

Una vez configurados los grupos de prácticas con la información y los alumnos correspondientes, serán convocados de manera periódica, típicamente semanal y alguna esporádica de recuperación, a una serie de sesiones de prácticas.

Cada grupo de prácticas debe ser capaz de reconocer y gestionar sus propias sesiones de prácticas y acceder y presentar la información de asistencia de la misma, si la hubiera. Las acciones que un tutor puede requerir de una sesión de prácticas específica son:

- **Ver sesión:** muestra los datos de asistencia de la sesión actual y permite modificarlo por cada alumno.

- Si la sesión de prácticas está activa, los alumnos podrán utilizar su tarjeta NFC para identificarse y confirmar su entrada o salida a la sesión.
- **Subir sesión:** confirma los datos de asistencia en docencia virtual.
  - Los datos de cada sesión se añadirán a un fichero denominado AsistenciaPracticas.csv, en el mismo directorio desde el que se descargó el fichero alumnos.csv para la asignatura actual. Si dicho fichero no existe, la aplicación deberá crearlo automáticamente.
  - Una vez que los datos de asistencia a la sesión estén confirmados en Ilias, no tendrá sentido ninguna modificación, edición de los datos de la sesión o su borrado, por lo que se deshabilitarán estas opciones.
- **Editar sesión:** permite modificar los datos de configuración de la sesión actual, siempre que no haya sido confirmada en docencia virtual.
- **Borrar sesión:** permite borrar una sesión actual, tanto su registro de la base de datos, como su fichero local con los datos de asistencia, siempre que dicha sesión no haya sido confirmada en Ilias.

Si el grupo no tuviera ninguna sesión de prácticas aún, se va a poder lanzar la acción de **crear nueva sesión de prácticas** desde la misma actividad.

Atendiendo a esto, el diagrama de casos de uso de esta actividad sería:

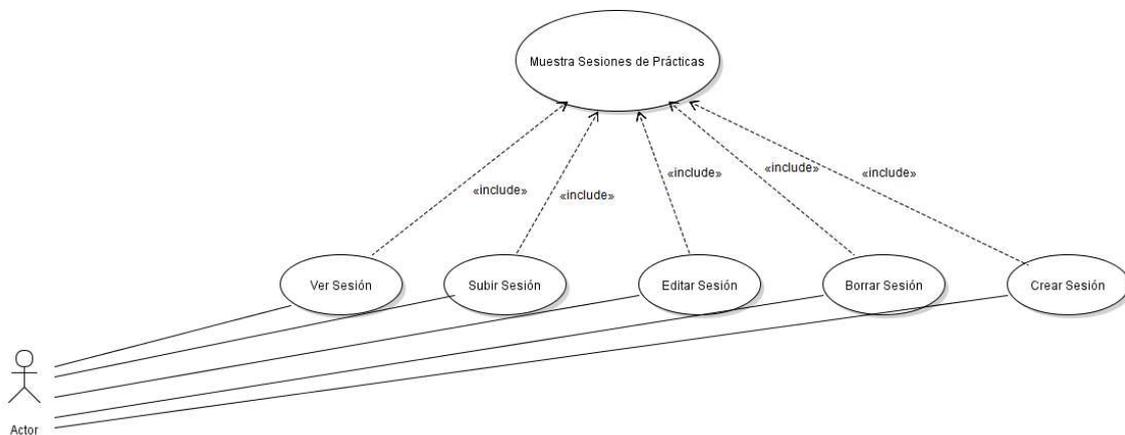


Figura 26 - Diagrama de Casos de Uso de Seleccionar Sesión

Y su diagrama de flujo sería:

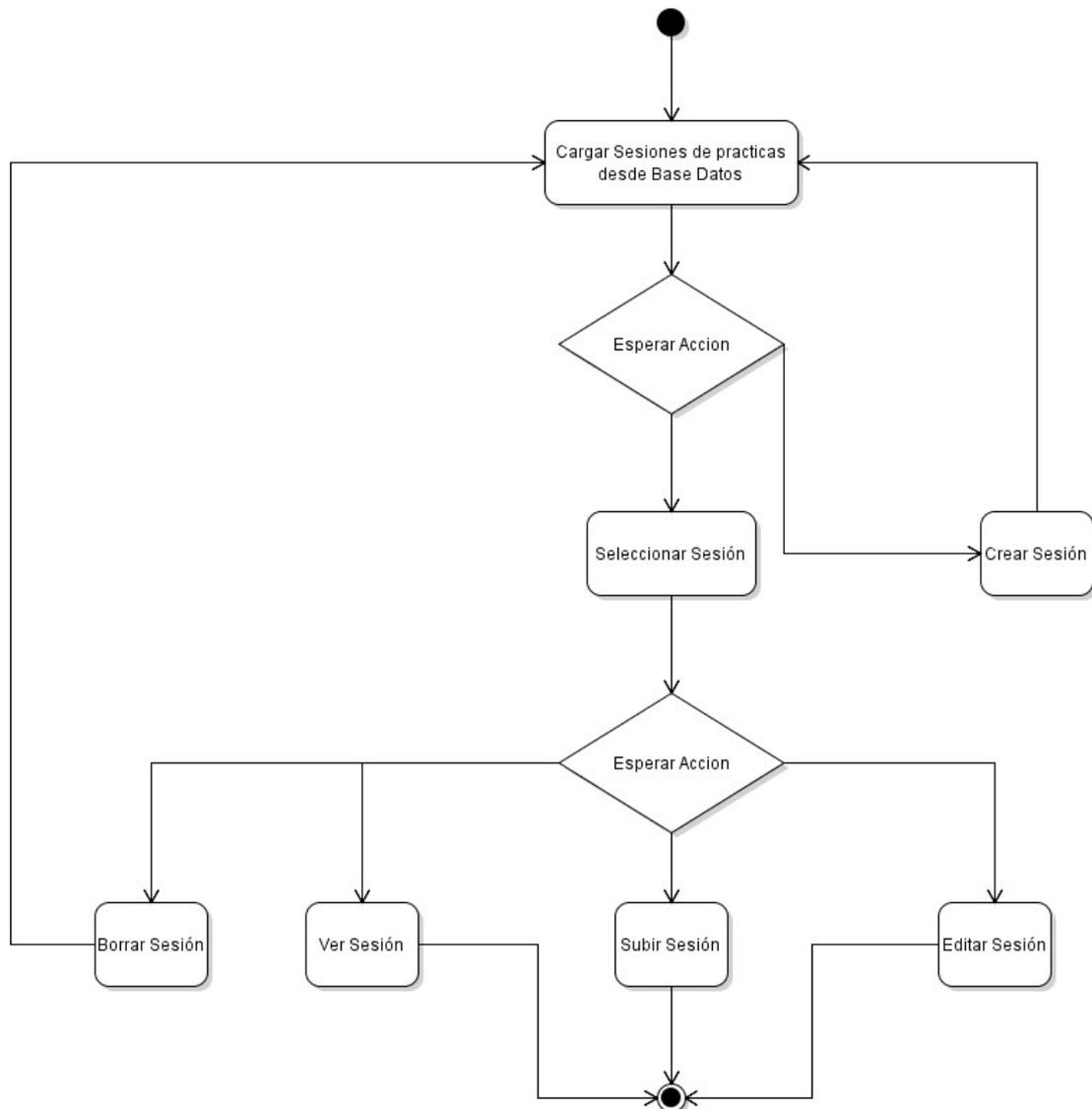


Figura 27 - Diagrama de Flujo de Seleccionar Sesión

Al ser un diagrama muy similar al mostrado en la Figura 16 - Diagrama Flujo Seleccionar Asignatura / Grupo Prácticas, se ha optado por desarrollar una actividad Android que mantenga el mismo patrón de diseño mostrado en la

Figura 18 – Asignatura con grupos, con una lista desplegable en la que se carguen todas las sesiones y, cada vez que se pulse sobre cualquiera de ellas, se muestre un menú con las opciones: Ver Sesión, Subir Sesión, Editar Sesión y Borrar Sesión, junto con la barra de acción que incluya la opción de Crear Sesión a la vista. Por cada sesión de prácticas, la información que se mostrará será la fecha y hora en la que está convocada la misma y si los datos de asistencia han sido subidos a Ilias o no. Se usará el siguiente diseño:

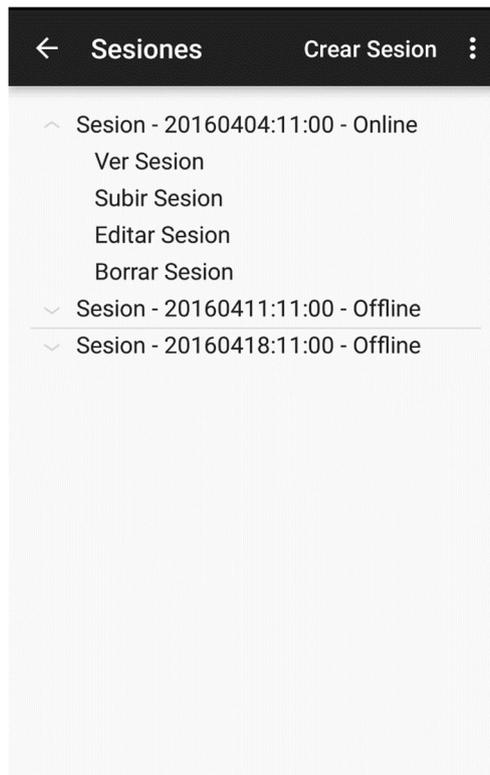


Figura 28 - Seleccionar Sesión de Prácticas

#### 1.2.3.9 Crear Sesión de Prácticas

Cuando se crea un grupo de prácticas, no se generan sus sesiones de forma automática, ya que se desconoce la fecha inicial en la que empiezan a contar las sesiones y no se dispone en docencia virtual de información sobre la duración de la asignatura: cuatrimestral, semestral, anual, etc., como para calcular la fecha de la última sesión.

Es por esto, por lo que se provee a la aplicación de una actividad adicional que permite crear sesiones de prácticas. Dicha actividad permitirá definir la fecha y la hora de la sesión y su duración estimada para cada sesión concreta. Para facilitar la tarea de crear las sesiones de un periodo de tiempo concreto, se permite introducir una fecha de fin y, atendiendo a esto, la aplicación creará una sesión semanal desde el día de origen hasta el día de fin, siempre a la hora indicada. Hay que prestar especial atención aquí a no repetir sesiones en fecha y hora, puesto que no sería válido.

Este es el diagrama de flujo de la actividad:

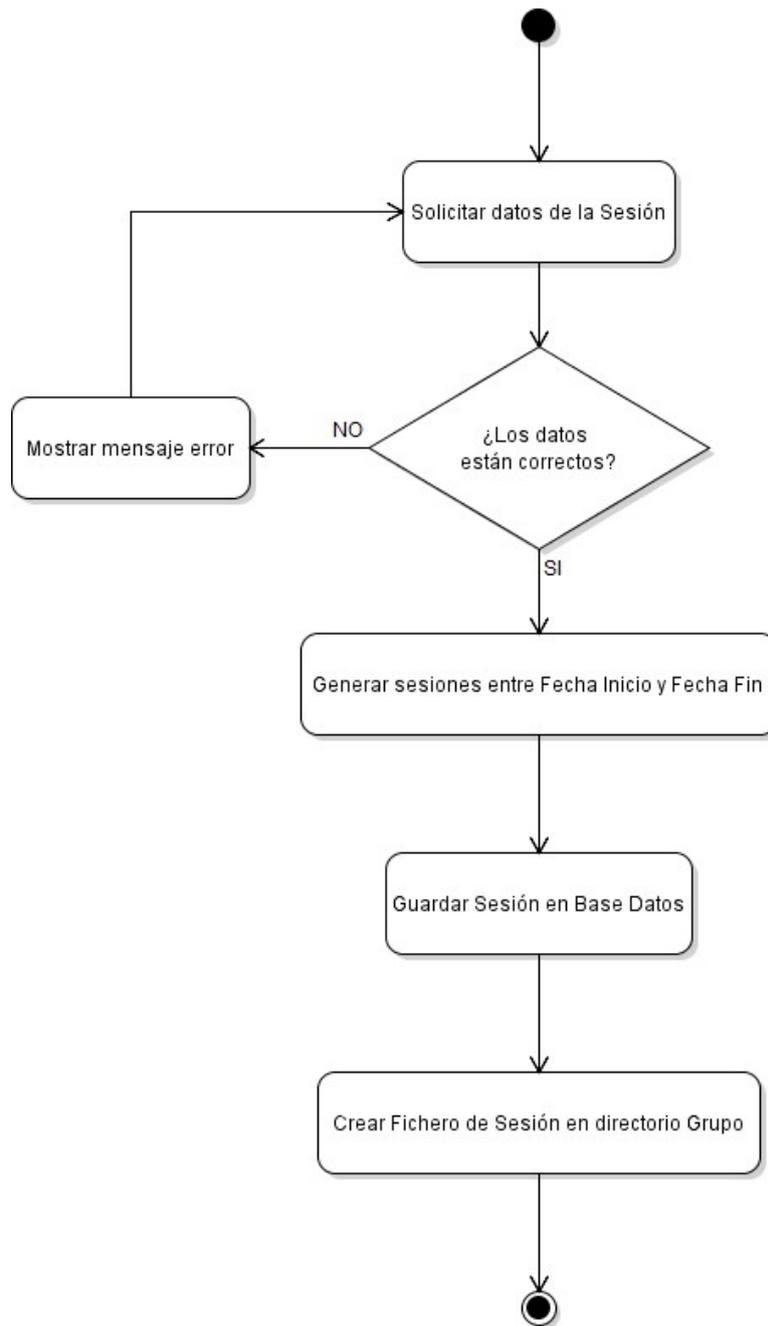


Figura 29 - Diagrama Flujo Crear Sesión

El diseño de la misma consiste en un formulario simple para recoger los datos de la sesión:

← Nueva Sesión

Crear Sesión  
Introduzca fecha/hora prevista de sesión

Fecha

Hora  Duración (minutos)

Marcar para crear un conjunto de sesiones

Fecha Fin

Aceptar

Figura 30 - Crear Sesión de Prácticas

En él, se aprecia que el cuadro de texto que solicita la fecha de fin está desactivado por defecto. Esto es debido a los comentarios de los usuarios que testearon la primera beta de la aplicación, que encontraban más intuitivo generar una sesión por defecto y activar la generación de sesiones en batería como una opción. Analizando la propuesta se llega a la conclusión de que como la generación de sesiones en batería se hará una vez por asignatura; la opción más usada a lo largo del año será la creación de sesiones individuales (recuperaciones, sesiones especiales, ...), por lo que se aceptó utilizar este comportamiento como comportamiento por defecto.

Una vez que la sesión de prácticas ha sido almacenada, hay que crear el fichero de sesión correspondiente en la carpeta del grupo para el que se genera la sesión, con el formato definido en el apartado *Modelo de Datos*. Dicho fichero se identificará con el id de sesión, su fecha y su hora, por lo que no se admiten duplicados de fecha y hora:

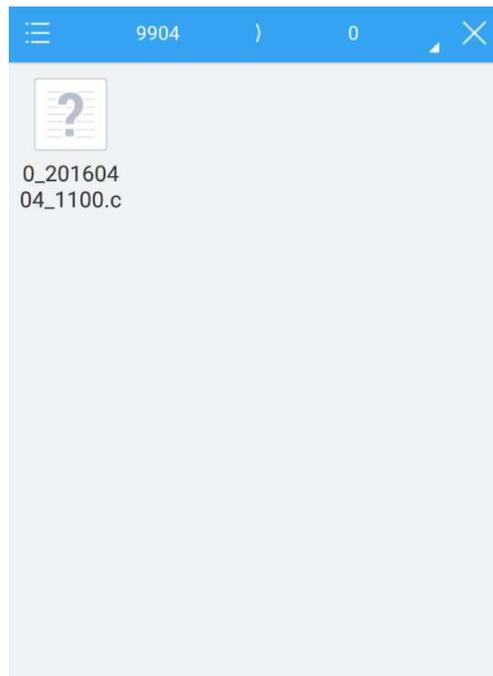


Figura 31 - Fichero de Sesión

#### 1.2.3.10 Editar Sesión de Prácticas

Partiendo del estado en que un determinado grupo de prácticas ya tiene una o más sesiones definidas, es factible que el usuario necesite realizar operaciones de edición sobre las mismas. Se contemplan dos tipos de operaciones: **editar los datos** que la definen y **borrar la sesión**, ambas seleccionables en la lista de opciones que se despliega al seleccionar una sesión cualquiera, como puede verse en la Figura 28 - Seleccionar Sesión de Prácticas. Por tanto, los casos de uso de esta actividad serían:

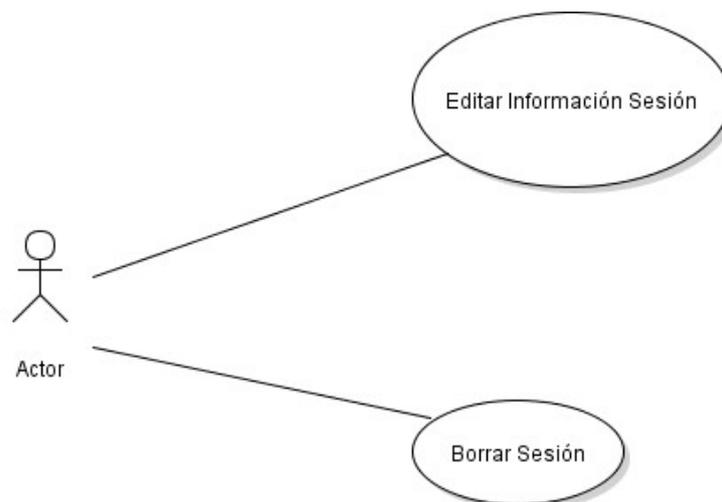


Figura 32 - Diagrama de Casos de Uso de Editar Sesión

Si el usuario decide editar la información de la sesión, se mostrará la misma actividad descrita en el epígrafe Crear Sesión de Prácticas, con la información de la sesión actual cargada en los campos correspondientes.

← Editar Sesión

Editar Sesión  
Introduzca fecha/hora prevista de sesion

Fecha 20160404

Hora	Duracion (minutos)
11:00	60

Aceptar

*Figura 33 - Editar Sesión*

Se puede observar que los campos correspondientes a la creación de las sesiones en batería están ocultos, para no confundir al usuario.

El diagrama de flujo asociado a esta acción sería el siguiente:

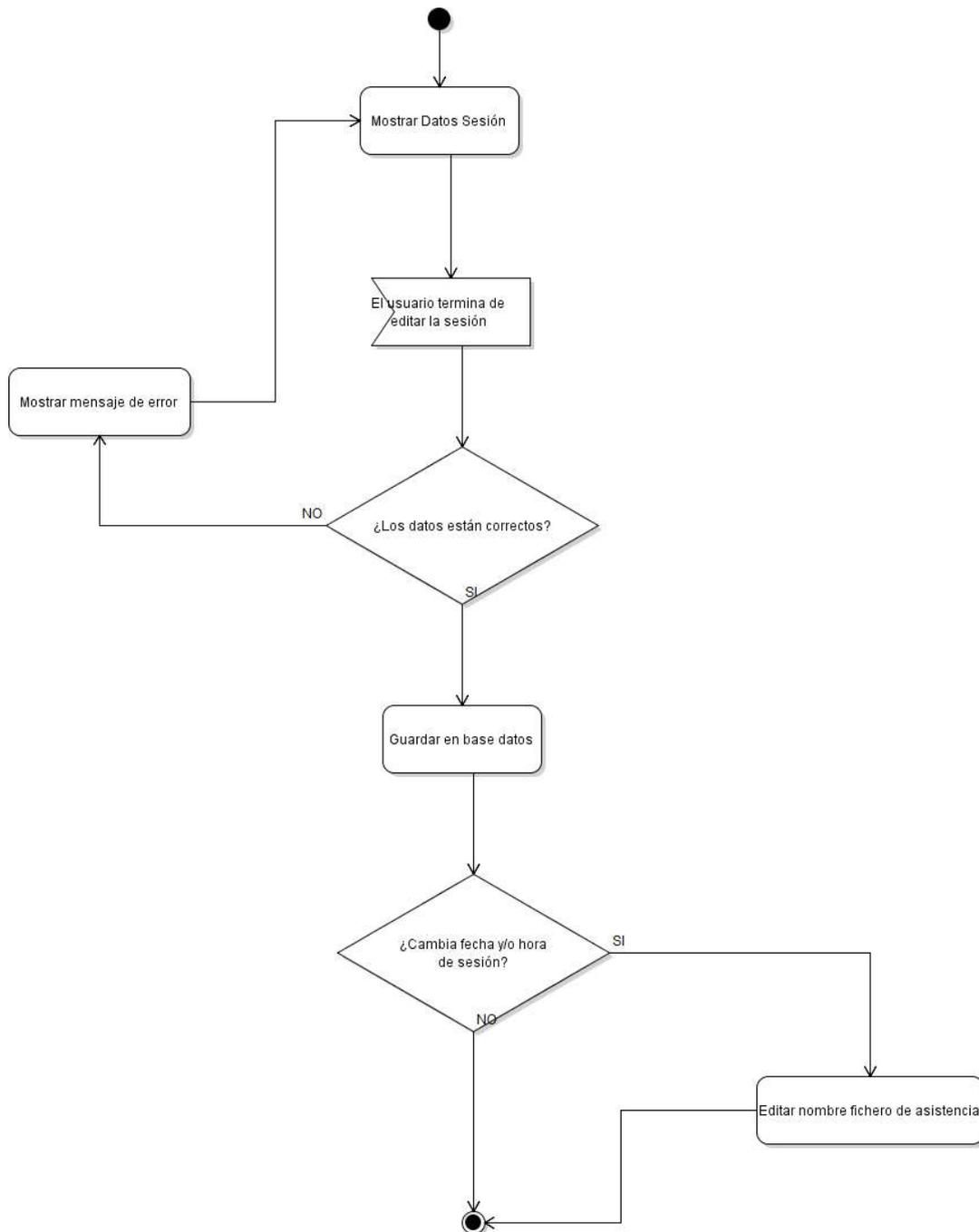


Figura 34 - Diagrama de flujo Editar Sesión

Si se modifica la fecha o la hora de la sesión, la aplicación editará automáticamente el nombre del fichero de asistencia asociado, para que mantenga coherencia con la sesión actual y evitar problemas de coherencia si más adelante se crea otra sesión con la fecha y hora que tenía la actual.

Si el tutor decide eliminar la sesión de prácticas, la aplicación mostrará un diálogo solicitando confirmación de la operación, ya que los datos quedarán eliminados definitivamente y, si se confirma el borrado se eliminará tanto el registro de la sesión de

la base de datos como su fichero de asistencia correspondiente, en un diagrama de flujo como el siguiente:

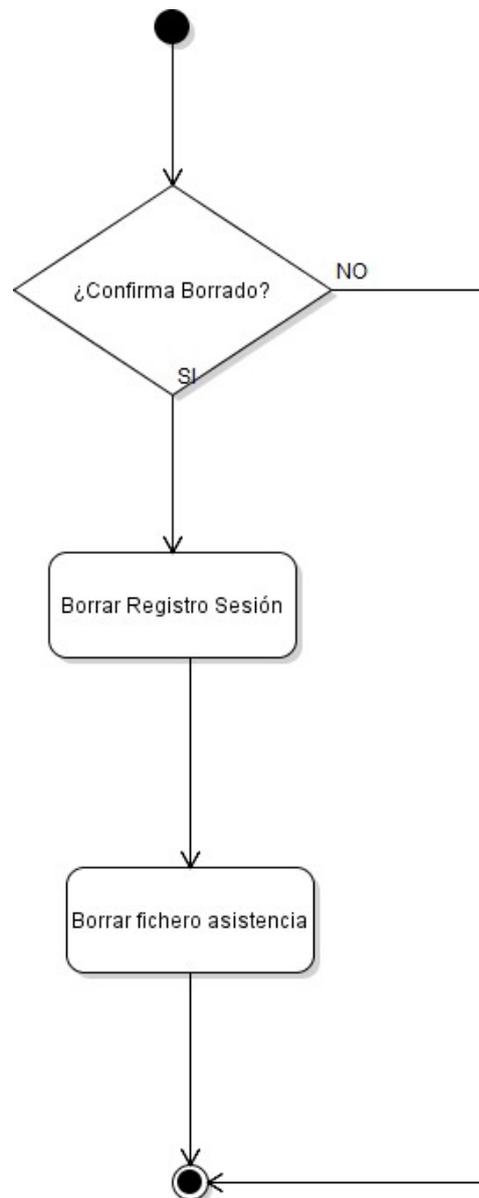


Figura 35 - Diagrama de Flujo Borrar Sesión

El diálogo de confirmación de borrado se mostrará utilizando la clase AlertDialog que implementa Android y se vería así:

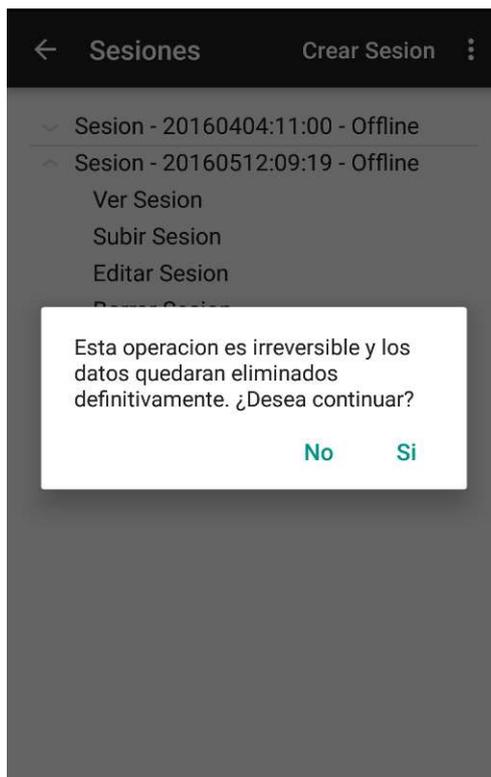


Figura 36 - Confirmación Borrado Sesión

Ninguna de estas opciones tendría sentido ejecutarse si los datos de asistencia ya han sido confirmados en Ilias, puesto que produciría falta de coherencia entre los datos. Hay dos formas de implementar esta característica:

- No mostrando las opciones: la implementación es prácticamente inmediata y el usuario percibe que puede realizar menos operaciones.
- Mostrando las opciones, pero rechazando la operación: la implementación también es sencilla; pero requiere código adicional para mostrar el mensaje que informe al usuario de la situación.

Se considera que el usuario será más consciente de las implicaciones que conlleva el subir una sesión si se muestran mensajes de “Operación Rechazada” en lugar de no mostrarle la opción, por lo que se opta por mostrar la opción y rechazar la operación.

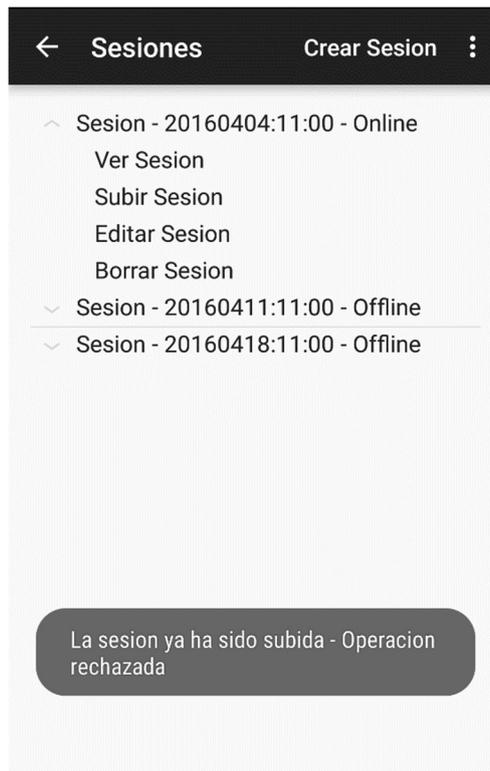


Figura 37 - Operación Sesión Rechazada

#### 1.2.3.11 Trabajar Sesión de Prácticas

Una vez que el tutor selecciona una sesión de prácticas, la aplicación debe mostrar una actividad en la que se carguen todos los alumnos del grupo y, por cada uno de ellos, se pueda anotar la entrada y la salida a la sesión, tal y como se solicita en el apartado *Objetivos - Toma de datos offline-actualización online, precarga de los datos de los alumnos de una asignatura antes de la sesión y registro de las entradas y salidas.*

Además, la aplicación debe ser capaz de leer las tarjetas NFC de los alumnos e identificarlos adecuadamente, para cumplir con el objetivo de *identificación del alumno a través de una tarjeta NFC personal.*

El diagrama de casos de uso de esta actividad sería:

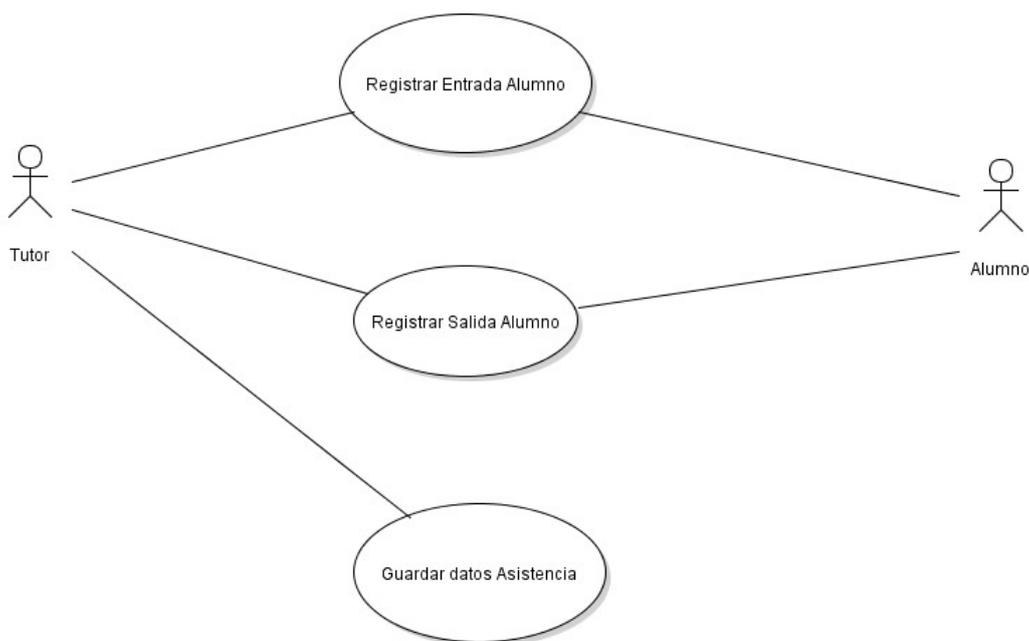


Figura 38 - Diagrama de Casos de Uso de Trabajar Sesión

La información que necesita añadir un tutor para poder registrar las entradas y salidas de un alumno a la sesión de prácticas son:

- Nombre completo.
- Cuadro de texto para introducir hora de entrada.
- Cuadro de texto para introducir hora de salida.

Analizando en detalle el formato que tendría la actividad asociada y el ritmo de entrada y salida de los alumnos, se observa que de tener que buscar e insertar por cada alumno que venga sin tarjeta NFC la hora de entrada y la hora de salida de forma manual, sería un lastre para la experiencia de usuario, por lo que se opta por registrar las horas de entrada y salida de forma automática, mediante una casilla de verificación que, una vez activada, lanzará una consulta a la hora del sistema para que se almacene en el registro de la sesión.

De igual forma que cuando se está trabajando con los alumnos de un grupo de prácticas (1.2.3.7), es posible que durante el transcurso de la sesión de prácticas se produzca alguna interrupción o evento durante la ejecución de la aplicación, bien sea porque el tutor necesita consultar otra aplicación, bien sea porque se recibe una llamada, etc; por lo que hay que diseñar un método que evite pérdidas de datos si se produce esta situación. Podría optarse por la misma solución que en el apartado 1.2.3.7 y guardar la información de entrada y salida cada vez que se registra una de estas acciones; pero podría darse que la aplicación lanzara una excepción no controlada, debido a que no se puede tener un acceso concurrente al fichero de asistencia de la sesión, provocado por el alto volumen de registros a escribir, ya que el sistema de almacenamiento sólido de Android es muy variado y el fichero podría estar incluso

ubicado en una tarjeta de memoria externa, con poca velocidad de escritura. Esto es un problema crítico, ya que haría que la sesión quedara corrupta y la invalidaría, por lo que hay que limitar al mínimo las operaciones de escritura al fichero de texto de asistencia.

Android dispone de una serie de métodos en su API, que se pueden sobrescribir para gestionar el ciclo de vida de una aplicación. De toda la variedad que implementa, a la aplicación le interesan dos de ellos para solucionar la casuística anterior: `OnPause()` y `OnResume()`. Ambos métodos se ejecutan por cada actividad individual que conforma la aplicación y su uso está recomendado para manejar este tipo de situaciones. `OnPause()` es el método que se lanza justo cuando se recibe cualquier interrupción de la actividad actual y permite salvar los datos de la misma, para que luego puedan ser cargados de nuevo en `OnResume()`. Se puede aprovechar este comportamiento para lanzar un salvado de la información cuando se lance `OnPause()` y que cuando se ejecute `OnResume()` se cargue toda la actividad como desde el principio, aprovechando que la información ya está guardada.

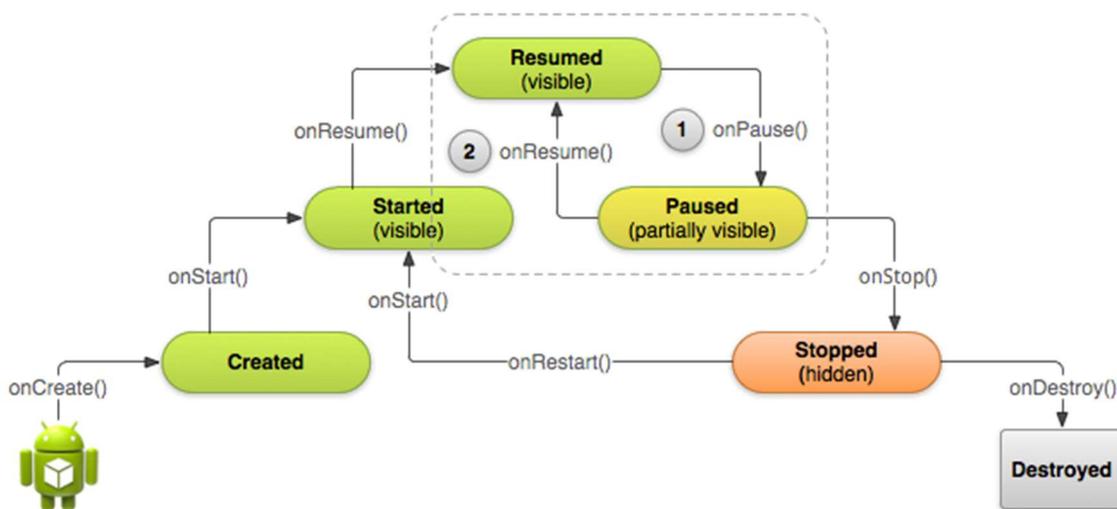


Figura 39 - Ciclo de vida de una aplicación Android

El diagrama de flujo para el correcto funcionamiento de la actividad sería:

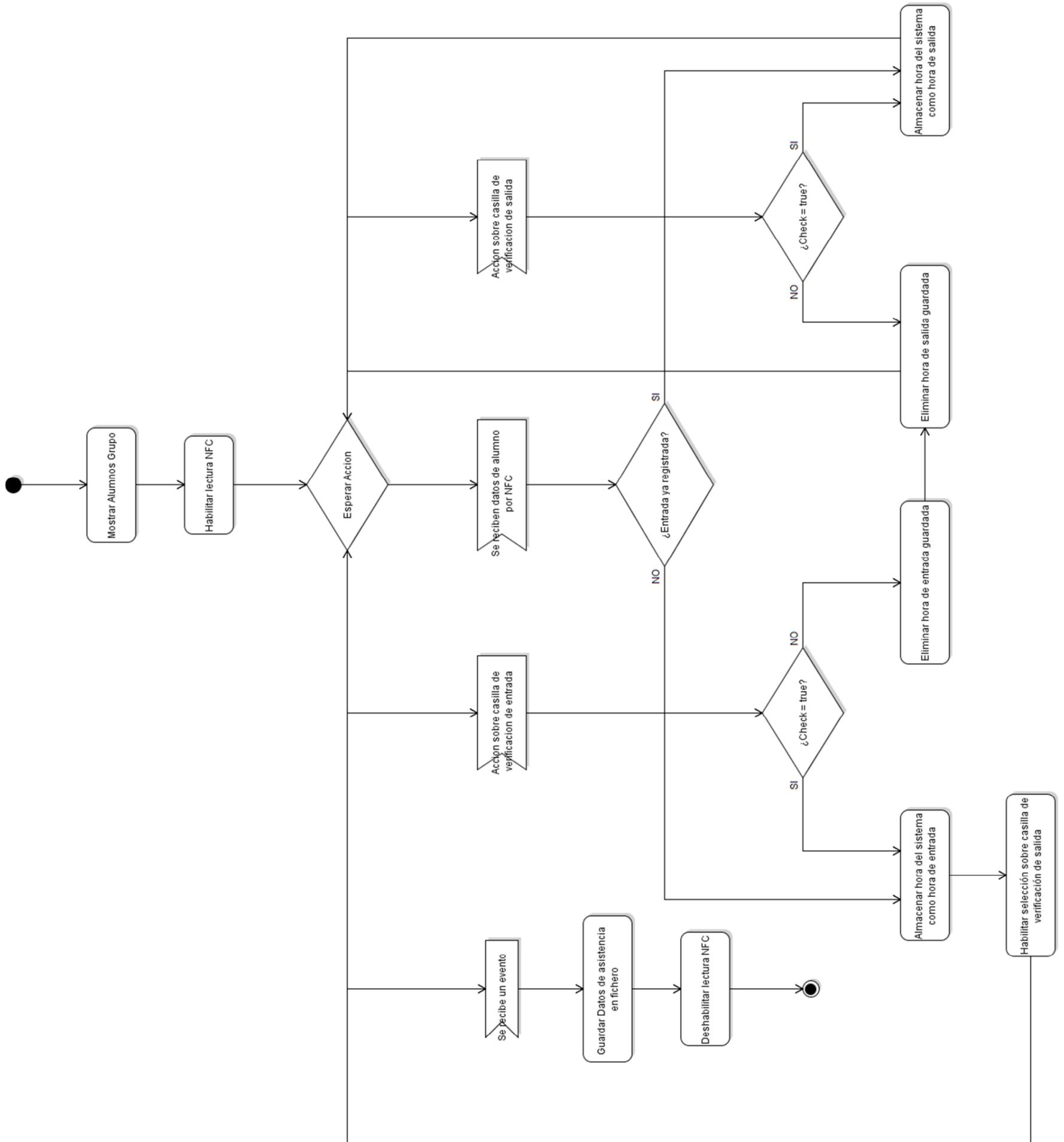


Figura 40 - Diagrama de flujo de Trabajar Sesión

Se puede observar en el diagrama de flujo que la lectura NFC solamente servirá para activar casillas de verificación: primero la de entrada y en una segunda lectura la de salida, puesto que no tendría sentido suponer que, si un alumno acerca su tarjeta, es para anular su asistencia.

Para habilitar a la aplicación a capturar y leer las tarjetas NFC que vayan acercando los alumnos, hay que activar el despachador de eventos de NfcAdapter (en la propia API de Android) durante la ejecución de **OnResume()** de la actividad y, para evitar errores posteriores, hay que desactivarlo en el **OnPause()**, ya que en otro caso, la lectura de una tarjeta NFC fuera de la actividad actual, haría que el sistema Android lanzara una instancia de la actividad actual de nuevo, pero sin datos de alumnos ni grupos, generando una excepción. Si todo está configurado correctamente, la captura de forma nativa se ejecuta por defecto en una actividad propia del sistema Android, por lo que se pierde el flujo de código dentro de la propia aplicación. Por este motivo la API Android utiliza en la llamada al despachador un objeto del tipo **PendingIntent**.

Un PendingIntent es una entidad que ejecuta código de una aplicación desde fuera de la misma, utilizando sus permisos e identidad, con lo cual se puede resolver el evento NFC y volver al punto actual de la actividad sin dificultad. Es tarea del programador, asegurarse de que la identidad de la actividad que lanza el PendingIntent es la correcta para evitar posibles problemas por suplantación de identidad. Aprovechando esta circunstancia, el PendingIntent va a intentar lanzar la actividad actual de nuevo. El problema de seguridad queda resuelto automáticamente, ya que la aplicación no tiene activa la captura de NFC fuera de la actividad actual, por lo que solamente se puede llegar aquí a través de un usuario correctamente identificado. Para garantizar una correcta experiencia de usuario, no sería correcto que hubiera múltiples instancias de la actividad actual y que cada vez que el usuario pulsara sobre el control "Back" de Android, el sistema volviera a la misma actividad, pero con un estado de asistencia anterior, ya que, además, como se puede ver en la Figura 40 - Diagrama de flujo de Trabajar Sesión, se guarda la asistencia cada vez que se realiza esta acción; por lo que también se estarían perdiendo registros de asistencia. Para evitar esto, se decide configurar el lanzamiento de la actividad actual como SINGLE\_TOP, es decir, solo puede haber una instancia de la actividad actual y se aprovechará el evento **OnNewIntent()**, que es el que se lanza cuando se necesita resolver crear una nueva actividad en modo SINGLE\_TOP, para comprobar que, efectivamente, se ha capturado una tarjeta NFC, leerla, buscar el DNI del alumno entre los alumnos de la asignatura y activar el registro de entrada/salida que corresponda.

La única forma de volver a lanzar la actividad actual desde ella misma es cuando se lee una tarjeta NFC, es necesario descartar el resto de posibles eventos durante la

ejecución de `OnNewIntent()`, por lo que habrá que recuperar la acción que dio lugar al `Intent` y, según la documentación de Android, si esta acción es `ACTION_NDEF_DISCOVERED`, `ACTION_TAG_DISCOVERED` o `ACTION_TECH_DISCOVERED`, es porque se ha leído una tarjeta NFC y, por tanto, se puede procesar la lectura de la misma.

Por otro lado, las sesiones pertenecen a un grupo de prácticas determinado; pero como se muestran todos los alumnos de las mismas, se ha decidido resaltar en color aquellos que no pertenecen al grupo de la sesión, de tal forma que tendremos:

- Sin resaltado: el alumno pertenece al grupo de la sesión.
- Rojo: el alumno pertenece a un grupo de prácticas distinto al de la sesión.
- Amarillo: el alumno no ha sido asignado aún a ningún grupo.

De esta forma se espera ayudar al tutor de forma visual a identificar cuáles son los alumnos que están recuperando asistencias o no tienen grupo asignado y poder así solucionar conflictos en las aulas como falta de puestos o grupos muy numerosos.

Además, la aplicación se ha diseñado para que el registro de entradas y salidas siga un flujo natural, de tal forma que no se permite registrar una salida si no ha habido una entrada previa. Y, si estando registrada ya la salida, el tutor decide invalidar la entrada, automáticamente se eliminarán del registro las horas de ambos eventos.



Figura 41 - Trabajar con Sesión

Cuando la sesión concluye y se cierra, la aplicación automáticamente establece la hora de salida de todos los alumnos que confirmaron su entrada y olvidaron confirmar

su salida, como la hora de fin teórica, según sus parámetros de hora de inicio y duración, indicados ambos durante la creación de la misma, con el objetivo de mejorar la experiencia de usuario.

En caso de que el fichero de sesión haya sido confirmado en Ilias, no se admitirá edición de los datos de asistencia para evitar falta de coherencia en los mismos. Sin embargo, sí se permitirá consultar la información por cualquier supuesto que se necesite, sin necesidad ni de acceder a Ilias, ni de trabajar con el fichero de texto de la sesión. Para ello, se presentará el mismo interfaz de la Figura 41 - Trabajar con Sesión; pero con cualquier opción de edición totalmente deshabilitada y la lectura de tarjeta NFC desactivada.



← Alumnos Sesion		
ZENOBIA HERRERO...	g1	<input checked="" type="checkbox"/> Entrada <input checked="" type="checkbox"/> Salida
LUIS CHAVES ARAN...	g1	<input checked="" type="checkbox"/> Entrada <input checked="" type="checkbox"/> Salida
ANTONIO MORENO...	g2	<input type="checkbox"/> Entrada <input type="checkbox"/> Salida
MARÍA DE LOS SA...	g2	<input type="checkbox"/> Entrada <input type="checkbox"/> Salida
DAVID BUSTAMANT...	Sin Grupo	<input checked="" type="checkbox"/> Entrada <input checked="" type="checkbox"/> Salida

Figura 42 - Trabajar con Sesión Confirmada

#### 1.2.3.12 Subir Asistencia a Ilias

El último objetivo que falta por cubrir es el de *guardar en Ilias en un formato fácilmente editable una vez se ha terminado la sesión*. Esta acción se puede realizar de forma automática cuando se cierra la sesión o de forma manual por el tutor. La gestión de la subida automática cuando termina la sesión genera el problema de falta de control del tutor de los datos a subir, ya que no se permitiría una revisión formal para confirmar la información. Por este motivo, se ha decidido implementar la subida de los datos manual, a partir de una orden del tutor.

Esta acción comienza cuando el tutor pulsa sobre la opción de “Subir Sesión” en la actividad de la Figura 28 - Seleccionar Sesión de Prácticas, y ejecuta el siguiente flujo:

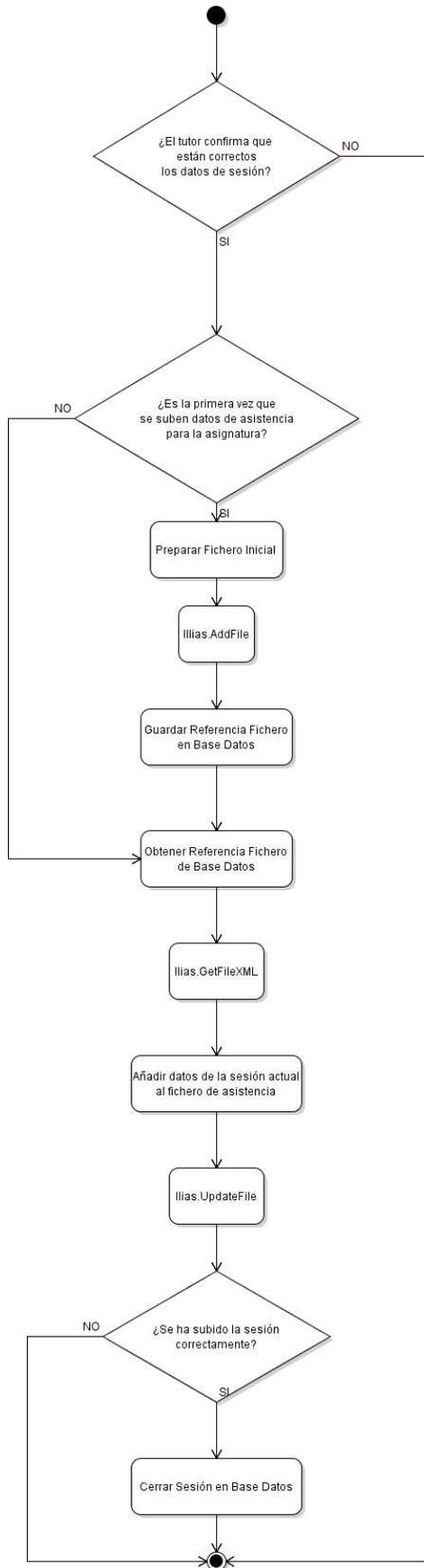


Figura 43 - Diagrama Flujo Subir Sesión

Para facilitar el flujo de la aplicación se ha decidido que la primera vez que se sube el fichero de asistencia, la aplicación hará dos operaciones en Ilias: en la primera se generará un fichero vacío con únicamente los nombres de las columnas que lo conforman, tal y como se definieron en el apartado de Fichero de sesión y, en la otra se descargará el fichero y se le añadirán los datos de asistencia, tal cual están en el fichero local. Esta segunda operación será común al resto de sesiones que se suban a partir de la primera, con lo que el código resultante es más sencillo de mantener.



Figura 44 - Fichero Asistencia cargado

La referencia al fichero de asistencia de cada asignatura se almacena en la base de datos cuando éste se crea, utilizando el parámetro devuelto en la llamada al servicio web AddFile.

Si el tutor accede al fichero a través de la propia plataforma de docencia virtual, podrá comprobar que el fichero se puede abrir con cualquier editor de texto e, incluso, con aplicaciones de hojas de cálculo.

	A	B	C	D	E	F	G	H	I	J	K
1	GrupoSesion	Sesion	Fecha	Hora	GrupoAlumno	IdAlumno	DniAlumno	Asistencia	HoraEntrada	HoraSalida	
2		0	1	20160404	11:00	0	8 44039224C	true	12:09	12:09	
3		0	1	20160404	11:00	-1	7 75110729C	true	12:09	12:09	
4		0	1	20160404	11:00	0	5 75108888L	true	12:09	12:09	
5											
6											
7											
8											

Figura 45 - Fichero de sesión en hoja de cálculo

Es especialmente crítico que, si la sesión no se sube a Ilias correctamente, no se pase a estado cerrada en la base de datos, puesto que se perdería la coherencia de la información. Atendiendo a esto, la sesión solamente pasara a estado cerrada en la base de datos si se recibe confirmación positiva de Ilias de la recepción del fichero.

### 1.3 Conclusiones

Una vez concluido el desarrollo, se evalúa el grado de cumplimiento de los diferentes objetivos.

#### Objetivos principales:

- La información se debe guardar en Ilias en un formato fácilmente editable una vez se ha terminado la sesión, con registro de las entradas y salidas.
  - Se ha conseguido cargar los datos de asistencia de varias asignaturas diferentes en Ilias, tal y como se puede ver por ejemplo en la Figura 44 - Fichero Asistencia cargado.
  - El formato del fichero de asistencia es un fichero de texto fácilmente editable y contiene registros de entrada y salida, como se aprecia en la Figura 45 - Fichero de sesión en hoja de cálculo
- Hacer una precarga de los datos de los alumnos de una asignatura antes de la sesión (Datos: nombre completo y fotografía del alumno).
  - Debido a la problemática de los permisos de Ilias, comentada en el punto 1.2.1.1 y fuera del alcance de este desarrollo, no se ha podido cargar la foto del alumno a la hora de mostrar la sesión; pero el resto de información del alumno se muestra en cada sesión y es fácilmente identificable (Figura 41 - Trabajar con Sesión). Además, se ha añadido la identificación visual de la correspondencia entre el grupo de la sesión de prácticas y el del alumno mediante un sencillo código de colores. Por último, se ha mantenido el diseño que muestra la imagen del alumno, para minimizar el impacto de un cambio en una versión posterior, cuando se solucione el problema de los permisos.
- Toma de datos offline-actualización online.
  - Los datos de los alumnos y las asignaturas se toman vía online, a través de docencia virtual y los grupos y las sesiones se crean en modo offline.
- Seguridad. Se deberán observar las medidas de seguridad pertinentes para evitar casos de suplantación de personalidad, copia de tarjetas, intrusión en las comunicaciones, etc.
  - Se requiere que el usuario se identifique en la plataforma de docencia virtual para poder acceder a la aplicación.
  - Las comunicaciones se realizan mediante NFC pasivo que es una tecnología de muy corto alcance en la que el único elemento que dirige la comunicación es el terminal del tutor, por lo que se minimizan las interferencias.

- El tutor puede comprobar la identidad del alumno in situ, contrastando la información de la tarjeta con la que tiene en pantalla.
- A la plataforma de docencia virtual no se carga ningún dato sensible del alumno, para preservar su privacidad.
- Los alumnos se identificarán a través de una tarjeta NFC personal.
  - La aplicación es capaz de leer tarjetas NFC y gestionar las entradas y salidas de los alumnos en la sesión de prácticas, tal y como se ha descrito en el apartado 1.2.3.11.

**Objetivos opcionales:**

- Uso de la emulación de Contacless Card a través de NFC en Android.
  - No se ha implementado.
- Control de alumnos que realizan sesiones de prácticas en grupos distintos al que pertenecen.
  - Se ha concebido un sistema prácticamente transparente al tutor para poder identificar los alumnos que recuperan sesiones, basado en un fichero de sesión con información concisa y eficaz, detallado en el apartado 1.2.2.2.
  - El sistema ideado permite, además, que los alumnos sin grupo puedan realizar sus sesiones de prácticas sin que esta situación suponga un perjuicio.

La aplicación ha sido probada en varios terminales Android, mostrando para cada usuario las asignaturas a las que tiene privilegios de acceso y con un funcionamiento correcto en todos ellos.

## 1.4 Líneas de Futuro

La utilidad de la aplicación para registrar la asistencia de los alumnos en sesiones de prácticas queda implementada. No obstante, a AsistenciaUJA se le podría dar continuidad tanto a corto plazo como a largo plazo de la siguiente forma:

- Corto plazo:
  - o Una vez que se rectifiquen los permisos de acceso a Ilias y los servicios web sean capaces de devolver los datos de los alumnos (cuando se realice una consulta con los suficientes privilegios), se debería eliminar la descarga de los alumnos a través del fichero alumnos.csv, previamente generado por el tutor y hacer todo el proceso a través de la plataforma. De esta forma se elimina un paso intermedio, susceptible a errores y se pasa a un proceso completamente automatizado.
- Largo plazo:
  - o Además de la asistencia a prácticas, las asignaturas pueden necesitar registrar la asistencia a clases de teoría e incluso, seminarios. Se puede evolucionar la base de datos para que distinga entre cada una y genere ficheros de asistencia para cada uno de los supuestos por separado.
  - o Otra opción interesante sería la sincronización de los grupos y las sesiones en Ilias. Esto permitiría que, si una asignatura tiene varios tutores, se puedan sincronizar los grupos entre los mismos, habilitando descarga automática de los datos o bajo petición. De esta forma cualquier tutor podría hacer una suplencia a otro sin perder la funcionalidad de AsistenciaUJA.

## 1.5 Anexos

### 1.5.1 Manual de Usuario

A continuación, se describen las acciones necesarias para que el tutor pueda manejar con éxito la aplicación.

#### 1.5.1.1 Instalación

Para instalar la aplicación, hay que comprobar que esté habilitada la opción de instalar aplicaciones de orígenes desconocidos en el menú de Ajustes > Seguridad, ya que la aplicación no está actualmente publicada en la tienda oficial de Google y, en otro caso, no podrá instalarse.

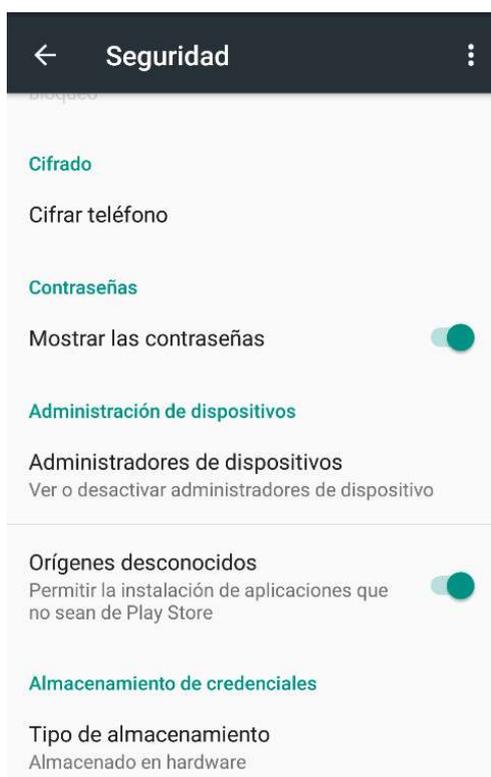


Figura 46 - Activar Orígenes Desconocidos

Una vez activada la opción, bastará con transferir la aplicación al dispositivo móvil y ejecutar el fichero AsistenciaUJA.apk para instalarla.

#### 1.5.1.2 Inicio

Para iniciar la aplicación, habrá que pulsar directamente sobre el icono de la misma.



Figura 47 - Icono AsistenciaUJA

Todas las actividades de la aplicación contienen una ayuda implementada para su consulta en tiempo de ejecución. Para mostrarla, bastará con pulsar sobre el icono de ajustes de cada actividad (representado por tres puntos en Android) y seleccionar la opción de ayuda, la cual mostrará una ventana de diálogo con información útil de la actividad actual en formato HTML para facilitar la navegación.



Figura 48 - Diálogo de Ayuda

### 1.5.1.3 Login

En la actividad de login se requiere que el usuario inserte sus credenciales de ingreso a la plataforma virtual de la Universidad de Jaén y pulse sobre el botón aceptar.

AsistenciaUJA

Login

Usuario

Usuario

Clave

Clave

Aceptar

*Figura 49 - Manual Usuario Login*

Si el usuario ha introducido correctamente el par usuario – clave y el teléfono tiene conectividad con internet, se pasará a la actividad Seleccionar Asignatura / Grupo de Prácticas.

Si el usuario pulsa en el menú de ajustes de la actividad, representado por tres puntos, puede desplegar el diálogo AcercaDe, para mostrar datos generales de la aplicación, así como su autoría:



Figura 50 - Diálogo Acerca De

#### 1.5.1.4 Seleccionar Asignatura / Grupo de Prácticas

En esta actividad se selecciona la asignatura con la que se desea trabajar en el panel deslizante (Imagen Izquierda). Una vez seleccionada aparecerá una lista con todos los grupos de prácticas ya generados en formato expandible, de tal forma que al pulsar sobre ellos se mostrarán las operaciones posibles para dicho grupo (Imagen derecha).



Figura 51 - Manual Usuario Seleccionar Asignatura / Grupo Prácticas

Si se desea seleccionar una asignatura diferente bastará con deslizar el dedo desde el extremo izquierdo de la pantalla hacia la zona central, para volver a desplegar la lista de asignaturas.

- Panel de asignaturas: contiene una lista de las asignaturas a las que tiene acceso el usuario de Ilias introducido en el paso anterior.
- Lista de grupos: contiene los grupos de prácticas que se han creado para la asignatura seleccionada. Por cada grupo se muestra la siguiente información:
  - o [Identificador] [(id interno)] – [Día de la semana] [Hora] – [Total Alumnos del grupo]

Permite realizar las siguientes acciones:

- o Editar Grupo: permite modificar los parámetros del grupo.
- o Ver Alumnos: permite comprobar y modificar los alumnos pertenecientes al grupo actual.
- o Ver Sesiones: permite comprobar y modificar las sesiones pertenecientes al grupo actual.
- o Crear Grupo: permite definir un nuevo grupo que será añadido a la lista.
- o Resetear datos: permite reiniciar los datos de la aplicación, perdiendo toda la información existente. Se recomienda su uso para cambiar de año lectivo.
- o Ayuda: documentación de ayuda asociada a la actividad.

#### **1.5.1.5 Crear Grupo de Prácticas**

En la siguiente actividad se permite crear un nuevo grupo de prácticas:

Figura 52 - Manual Usuario Crear Grupo Prácticas

Los distintos campos tienen el siguiente significado:

- Identificador de grupo: permite definir un identificador para el grupo que aparecerá en la lista de grupos.
- Descripción del grupo: permite introducir información adicional para el grupo actual, que no será mostrada en la lista de grupos. Solo se podrá consultar en esta actividad.
- Día de la semana: permite indicar el día de la semana en la que el grupo tiene la sesión normalmente, aunque no es vinculante para la creación de sesiones. Al hacer doble clic, se muestra un selector de fecha para indicar el dato de forma más cómoda.
- Hora: permite indicar la hora en la que el grupo tiene la sesión normalmente, aunque no es vinculante para la creación de sesiones. Al hacer doble clic, se muestra un selector de hora para indicar el dato de forma más cómoda.
- Aceptar: una vez que se ha cumplimentado la información requerida del grupo, hay que pulsar en aceptar para confirmar los datos.
- Ayuda: documentación de ayuda asociada a la actividad.

#### 1.5.1.6 Alumnos del Grupo de Prácticas

Esta actividad permite configurar los alumnos que pertenecen al grupo de prácticas seleccionado en el paso 1.5.1.4:

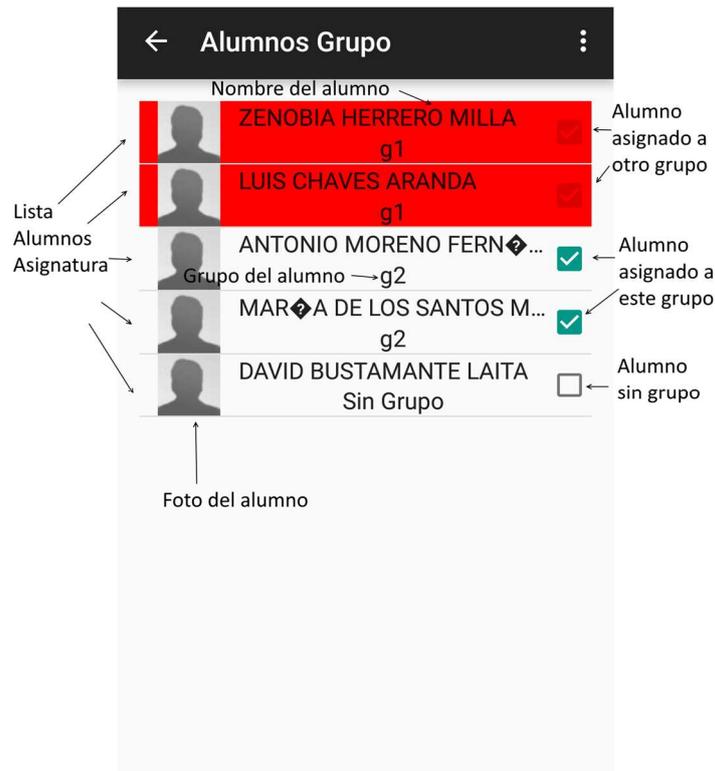


Figura 53 - Manual Usuario Alumnos Grupo

Se muestra una lista con todos los alumnos de la asignatura actual y el grupo al que pertenecen.

- Si el alumno está "Sin Grupo", se podrá asociar al grupo actual activando en el checkbox de su fila.
- Si el alumno pertenece al grupo actual, se podrá desasociar del grupo actual y dejarlo "Sin Grupo" desactivando el checkbox de su fila.
- Si el alumno está asociado a otro grupo, estará destacado en rojo y no se podrá asociar al grupo actual, ya que el checkbox de su fila estará deshabilitado.
- Ayuda: documentación de ayuda asociada a la actividad.

#### 1.5.1.7 Seleccionar Sesión de Prácticas

En esta actividad se muestra una lista con las sesiones de prácticas generadas para el grupo actual, en formato expandible, de tal forma que al pulsar sobre ellas se mostrarán las operaciones posibles para dicha sesión.

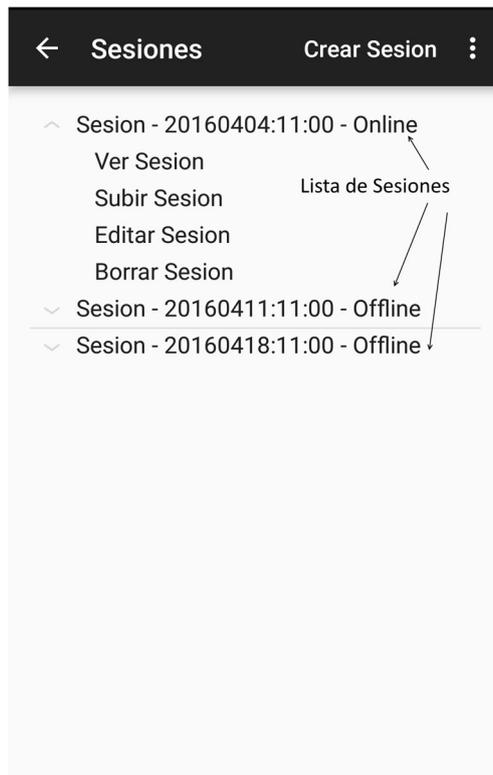


Figura 54 - Manual Usuario Seleccionar Sesión de Prácticas

Esta es la utilidad de los distintos elementos de la actividad:

- Lista de sesiones: contiene las sesiones de prácticas del grupo seleccionado. Por cada sesión se muestra la siguiente información:
  - o Sesion – [Fecha] [Hora] – [Estado Online]
- Permite realizar las siguientes acciones:
  - o Ver Sesión: permite comprobar y establecer (únicamente si la sesión está en estado Offline) la entrada y salida de alumnos de la sesión actual.
  - o Subir Sesión: permite confirmar los datos de asistencia en Ilias. Una vez realizada esta operación, el [Estado Online] de la sesión pasará de ser Offline a Online.
  - o Editar Sesión: permite modificar los parámetros de la sesión si la sesión está en estado Offline.
  - o Borrar Sesión: permite eliminar la sesión actual y sus datos asociados si la sesión está en estado Offline.
  - o Crear Sesión: permite definir una nueva sesión que será añadida a la lista.
  - o Ayuda: documentación de ayuda asociada a la actividad.

#### 1.5.1.8 Crear Sesión de Prácticas

En la siguiente actividad se permite crear una nueva sesión de prácticas:

Figura 55 - Manual Usuario Crear Sesión Prácticas

Los distintos campos tienen el siguiente significado:

- Fecha: permite indicar la fecha en la que se celebrará la sesión de prácticas y que puede ser distinto al indicado en el grupo. Al hacer doble clic, se muestra un selector de fecha para indicar el dato de forma más cómoda.
- Hora: permite indicar la hora en la que se celebrará la sesión de prácticas y que puede ser distinta a la indicada en el grupo, aunque por defecto aparece ésta. Al hacer doble clic, se muestra un selector de hora para indicar el dato de forma más cómoda.
- Batería de sesiones: si se activa el checkbox, se habilita un cuadro de texto para introducir una fecha de fin (con selector activable mediante doble clic), con el objetivo de crear una batería de sesiones desde la fecha indicada al principio hasta la fecha de fin introducida. Se generará una sesión por semana, por lo que se recomienda respetar el día de sesión del grupo cuando se generen sesiones mediante este procedimiento.
- Aceptar: una vez que se ha cumplimentado la información requerida de la sesión, hay que pulsar en aceptar para confirmar los datos.
- Ayuda: documentación de ayuda asociada a la actividad.

#### 1.5.1.9 Alumnos de la Sesión de Prácticas

Esta actividad permite establecer la asistencia de los alumnos a la sesión de prácticas seleccionada en el paso 1.5.1.7:

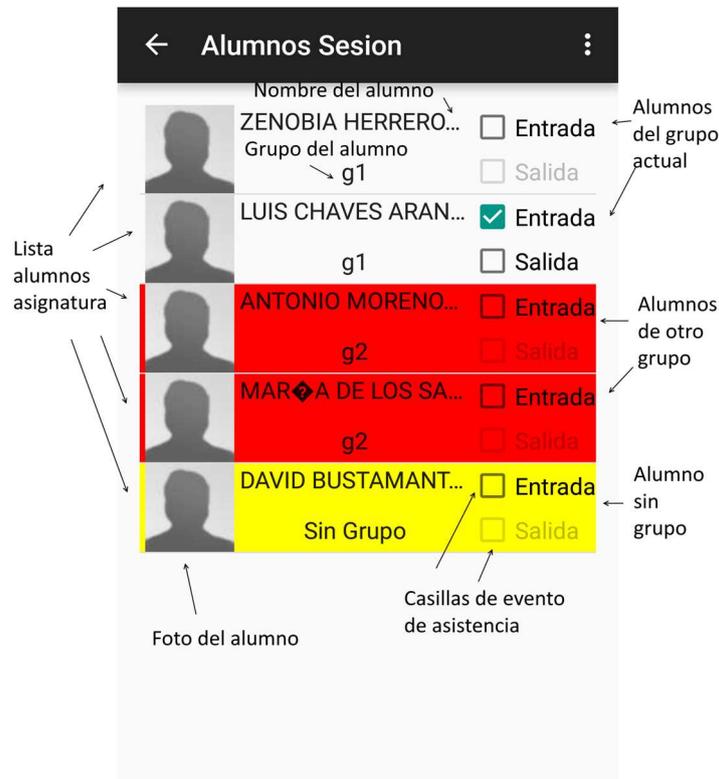


Figura 56 - Manual Usuario Alumnos Sesión

- Código de colores: se muestra una lista con todos los alumnos de la asignatura actual y el grupo al que pertenecen.
  - o Si el alumno está "Sin Grupo", se mostrará resaltado en amarillo.
  - o Si el alumno pertenece al grupo actual, se mostrará sin resalto de color.
  - o Si el alumno está asociado a otro grupo, estará destacado en rojo.

Al mostrarse todos los alumnos de la asignatura, se permite recuperar sesiones anteriores a alumnos que no pudieron asistir a su grupo original y que los alumnos sin grupo no pierdan asistencia.
- Registro de entradas y salidas: además, cada alumno tiene asociados dos checkboxes:
  - o Entrada: al activarse, queda registrada en el archivo de sesión en el almacenamiento interno del sistema, la hora a la que el alumno ha entrado a la sesión. Esta acción habilita el checkbox de salida para operar.
  - o Salida: una vez que la entrada ha sido activada, se habilita la posibilidad de registrar la hora de salida del alumno. Al activar el checkbox de salida, queda registrada en el archivo de sesión en el almacenamiento interno del sistema, la hora a la que el alumno ha abandonado la sesión. Si se cierra la actividad actual sin haber

registrado la hora de salida y la sesión ya ha expirado (de acuerdo a la hora de inicio y al tiempo de sesión configurado), se registrará automáticamente la hora de fin de la sesión para todos los alumnos.

- NFC: en esta actividad está activo el gestor de eventos NFC, de tal forma que si un alumno acerca una tarjeta NFC con el formato:
  - o DNI,NOMBRE,APELLIDO1,APELLIDO2

El sistema será capaz de registrar la actividad de entrada/salida que esté pendiente, es decir, si el alumno no ha registrado la entrada aún, al acercar la tarjeta NFC al dispositivo, se registrará la entrada; en cambio si la entrada ya ha sido capturada y se acerca la tarjeta, se capturará la salida. En caso de estar registradas ambas actividades, no se realizará ninguna acción.

### 1.5.2 Manual de Mantenimiento

En este apartado se asociarán las actividades indicadas en el manual de usuario con su código y su ubicación dentro del sistema, con el objetivo de facilitar la evolución de la aplicación y la corrección de errores.

La aplicación se entrega en un archivo anexo a la memoria, denominado AsistenciaUJA.zip. Este archivo contiene por un lado, el código de la misma y, por otro el fichero AsistenciaUJA.apk, listo para ser instalado tal como se describe en el proceso Instalación.

La aplicación está almacenada en el directorio AndroidStudioProjects creada por el propio AndroidStudio. La ubicación de dicho directorio se configura durante la instalación del entorno de desarrollo y, si se navega hasta él, se observa la carpeta AsistenciaUJA donde se guardarán los ficheros generado para la aplicación.

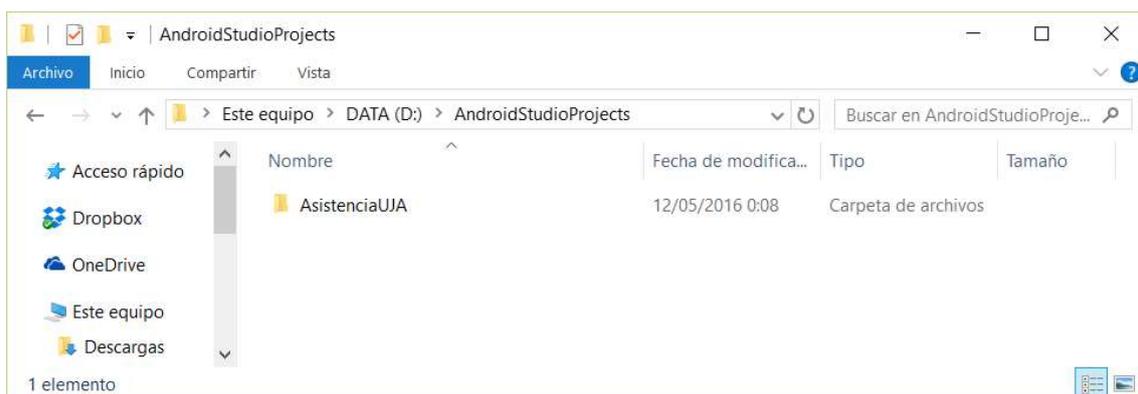


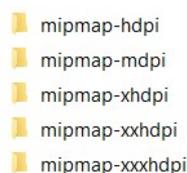
Figura 57 - Ubicación de la aplicación

Los ficheros que han sido específicamente desarrollados para la aplicación están dentro de AsistenciaUJA > app > src > main.

### 1.5.2.1 *AndroidManifest*

En el directorio main, se encuentra el fichero XML que define la configuración de cualquier aplicación Android, *AndroidManifest.xml*. De todo su contenido, se destaca lo siguiente:

- Espacio de nombres predeterminado: `com.uja.telematica`.
- Versión actual de la aplicación: `android:versionCode = "1"` y `android:versionName = "1.0"`.
- Versión mínima del SDK que se necesita para instalar la aplicación: `android:minSdkVersion="10"`, equivalente a Android 2.3.
- Versión del SDK utilizada para probar el desarrollo: `android:targetSdkVersion="21"`, equivalente a Android 5.0.
- Permisos utilizados por la aplicación: las aplicaciones Android necesitan que el desarrollador solicite explícitamente los permisos para utilizar determinados elementos del dispositivo, como el almacenamiento interno, el chip NFC o el acceso a internet. En AndroidStudio 2.0, los permisos se añaden de forma automática cuando se van utilizando las clases que los requieren.
- Icono de la aplicación: `android:icon="@mipmap/icono_launcher"`. Esto quiere decir que el icono se encuentra dentro del directorio `mipmap`, que debe existir dentro del actual. Normalmente todas las imágenes se incluyen en varios formatos para dar soporte a las distintas resoluciones y calidades de pantalla que se puede encontrar en toda la variedad de dispositivos en los que se puede instalar la aplicación. La resolución se indica con un sufijo en el nombre de la propia carpeta `mipmap`:



*Figura 58 - Resoluciones Icono Inicial*

Estos sufijos indican la resolución de la imagen con respecto al tamaño “medio” `mdpi`:

MDPI – Original (48 x 48).

HDPI - 1.5 x MDPI.

XHDPI - 2.0 x MDPI.

XXHDPI – 3.0 x MDPI.

XXXHDPI - 4.0 x MDPI.

- Nombre de la aplicación: `android:label="@string/app_name"`. Esta forma de representación, indica que hay que acceder a los recursos string, que se verán más adelante, para recuperar la clave `app_name`, con el nombre de la aplicación.
- Tema de la aplicación: contiene las normas de estilo generales, que serán respetadas en todas las actividades, `android:theme="@style/AppTheme"`. Esta nomenclatura se resuelve igual que la anterior.
- Actividad inicial de la aplicación: utilizada como punto de entrada a la misma. Debe estar categorizada como LAUNCHER, `<category android:name="android.intent.category.LAUNCHER" />`. En el caso de AsistenciaUJA es `android:name=".GUI.MainActivity"`.
- Actividades que la aplicación utiliza: además de la actividad inicial, también hay que indicar el resto de actividades que componen la aplicación.

### 1.5.2.2 Librerías Externas

En el caso de AsistenciaUJA, se utiliza la librería externa KSoap2 para soportar las comunicaciones de los servicios web a través del protocolo SOAP. Dicha librería está almacenada bajo el directorio AsistenciaUJA > app > lib.

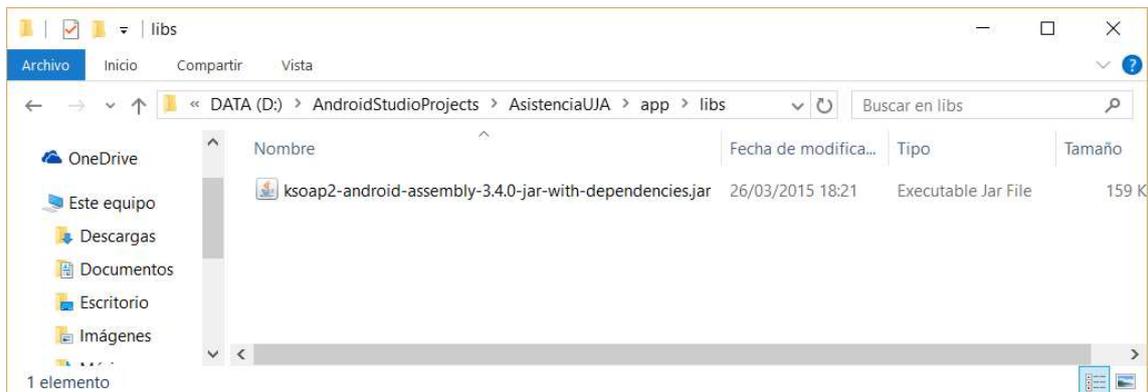


Figura 59 - Directorio Librería KSoap2

### 1.5.2.3 Recursos Internos

Las aplicaciones Android tienen una serie de recursos que han de ser declarados en tiempo de desarrollo y son reconocidos por el sistema para ser utilizados durante la fase de implementación, mediante las clases globales que provee Android. Estos recursos se almacenan bajo el directorio `res`, creado automáticamente por AndroidStudio.

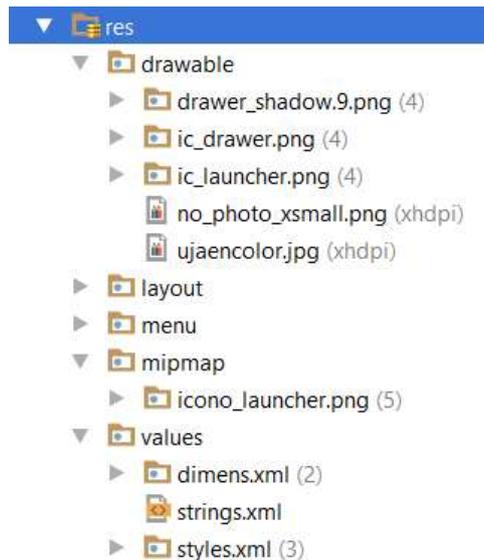


Figura 60 - Directorio res

Bajo este directorio se observan los siguientes elementos destacados:

- Drawable: Recursos que se utilizarán para ser dibujados en una o varias actividades Android. Se destacan los siguientes:
  - o Ic\_launcher: icono de la aplicación por defecto, en caso de no definirse ningún otro. En el caso de AsistenciaUJA existe por mipmap/icono\_launcher, por lo que éste no será utilizado.
  - o No\_photo\_xsmall: fotografía por defecto del alumno, cuando no se dispone de su imagen. Actualmente se utiliza para todos los alumnos.
  - o Ujaencolor: escudo de la Universidad de Jaén, mostrado en el diálogo Login > AcercaDe.
- Layout: contiene una serie de ficheros xml con el diseño de todas las actividades utilizadas por la aplicación.
- Menu: contiene una serie de ficheros xml con el diseño de las barras de acción que implementan todas las actividades. Típicamente en cada actividad mostrará su título, y una opción con tres puntos que presentará una ayuda y alguna opción extra, dependiendo de la actividad.
- Mipmap: contiene el icono de la aplicación. Se ha indicado en un directorio aparte para separarlo del resto de imágenes del sistema.
- Values: contiene una serie de ficheros xml con valores que podrán ser utilizados en el código.
  - o Dimens: contiene dimensiones empleadas en tiempo de diseño, como el ancho de la barra de navegación en la actividad Seleccionar Asignatura / Grupo Practicas.

- Strings: contiene todas las cadenas de texto utilizadas en la aplicación.
- Styles: contiene información del estilo visual de una actividad concreta o datos que afectan a toda la aplicación. Hay que destacar los siguientes valores globales:
  - Tamaño del texto por defecto: `<item name="android:textSize"> 17sp</item>`.
  - Estilo visual compartido: `<style name="AppBaseTheme" parent="Theme.AppCompat.Light">`.
  - Restricción de solo mayúsculas desactivada: para evitar que en la barra de acción sólo aparezca texto en mayúsculas, `<item name="textAllCaps"> false</item>`.

#### 1.5.2.4 Recursos Externos

Las aplicaciones Android tienen una serie de recursos que pueden ser utilizados en tiempo de ejecución y que no han de ser declarados en tiempo de desarrollo, sólo añadidos para su inclusión en la aplicación, con la contrapartida de que no son reconocidos por el sistema para ser utilizados durante la fase de implementación mediante las clases globales que provee Android. Estos recursos se almacenan bajo el directorio `assets`, que debe ser creado de forma manual por el desarrollador. En este él se almacenarán los ficheros de ayuda de cada actividad en formato HTML y que, posteriormente se mostrarán al usuario en una vista web bajo petición.

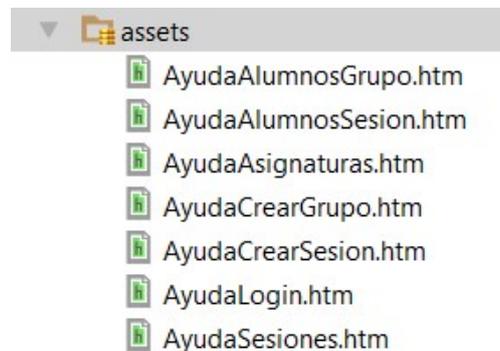


Figura 61 - Directorio `assets`

#### 1.5.2.5 Ficheros de Código

Los ficheros con la lógica de implementación se almacenan dentro del directorio Java de AndroidStudio. En él aparecen todos los espacios de nombres creados para la aplicación y los ficheros de código que los forman, organizados en carpetas creadas por el desarrollador:



Figura 62 - Directorio java

Actualmente se dispone de un espacio de nombres, que contiene tres directorios:

- GUI (*Graphic User Interface*): contiene un fichero de código por actividad, e implementa las acciones que realizan las mismas.
- DAO (*Data Access Objects*): contiene las clases necesarias para el acceso a elementos de datos tales como: Base de Datos, Ilias, fichero de asistencia de cada sesión y tarjeta NFC. Estos objetos serán utilizados por las actividades correspondientes del directorio GUI o los elementos de la clase BO, cuando se necesite la funcionalidad que implementan.
- BO (*Bussiness Objects*): contienen ficheros asociados a los elementos individuales de las listas, denominados adaptadores, utilizados para manejar las acciones presentadas en las actividades.

#### 1.5.2.6 Actividades

Por cada actividad descrita en el apartado *Manual de Usuario*, se va a presentar la correspondencia de todos sus ficheros, indicando para cada una: su descriptor de diseño en la carpeta layout, la correspondencia entre ese descriptor y su fichero de código en la carpeta GUI, el descriptor de diseño para su barra de acción y, si hubiera alguna lista, el adaptador de la carpeta BO asociado.

- Login: la actividad login es la única que está implementada a adrede utilizando el paradigma de fragmentos, que permite asignar vistas de diseño diferentes a una misma actividad, dependiendo de factores como la orientación del dispositivo, el tamaño de pantalla, etc. Se ha hecho así por el hecho de trabajar también con ellos de manera previa a las actividades que requieren de fragmentos de manera obligatoria.

Layout	<ul style="list-style-type: none"> <li>- activity_main: descriptor de diseño para el contenedor de fragmentos.</li> <li>- fragment_main: fragmento con el diseño de los componentes visuales.</li> </ul>
GUI	<ul style="list-style-type: none"> <li>- MainActivity: corresponde a la implementación de Activity_main.</li> </ul>

	- MainFragment: corresponde a la implementación de Fragment_main.
BO	
Menu	- Main.xml

Tabla 10 – Ubicación elementos Login

- Seleccionar Asignatura / Grupo de Prácticas: esta actividad contiene un panel de navegación para seleccionar asignaturas, que utiliza fragmentos generados automáticamente por AndroidStudio. Además, tiene una lista desplegable; pero, al ser solamente elementos de texto, no necesita de una vista de diseño personalizada por elemento, aunque sí necesite de un adaptador para manejar las opciones del mismo.

Layout	- asignaturas_fragment_navigation_drawer: contiene el diseño del panel de navegación. - activity_asignaturas: descriptor de diseño para el contenedor de fragmentos. - fragment_asignaturas: fragmento que contiene la lista de grupos. - opciones_lista_layout: descriptor para contener un único elemento de texto. Permite recuperar la opción seleccionada entre todas las presentadas por cada grupo de prácticas.
GUI	- AsignaturasFragmentNavigationDrawer: contiene la funcionalidad del panel de navegación. - AsignaturasActivity: funcionalidad de Activity_Asignaturas y Fragment_Asignaturas.
BO	- GrupoPracticasAdapter: adaptador para manejar las opciones desplegables por cada elemento de la lista de grupos.
Menu	- Asignaturas.xml

Tabla 11 – Ubicación elementos Seleccionar Asignatura / Grupo Prácticas

- Crear Grupo de Prácticas: esta actividad es un formulario simple, cuyos objetos tienen relación directa entre sí.

Layout	- activity_nuevo_grupo_practicas
GUI	- NuevoGrupoPracticas
BO	
Menu	- menu_nuevo_grupo_practicas

Tabla 12 – Ubicación elementos Crear Grupo

- Alumnos del Grupo de Prácticas: esta actividad implementa una lista que no es solo texto, por lo que se necesita un descriptor de diseño que se mostrará por cada elemento individual de la lista.

Layout	<ul style="list-style-type: none"> <li>- activity_alumnos_grupo: descriptor de diseño principal de la actividad. Contiene básicamente una lista en la que se mostrarán datos.</li> <li>- elemento_lista_grupo: contiene el diseño de cada elemento de la lista.</li> <li>- grupo_list_layout: descriptor que solamente contiene un checkbox. Permite recuperar el checkbox seleccionado de un elemento_lista_grupo.</li> </ul>
GUI	- AlumnosGrupoActivity: implementa la lógica de activity_alumnos_grupo.
BO	- AlumnoListAdapter: contiene el manejador de cada elemento de la lista elemento_lista_grupo.
Menu	- menu_alumnos_grupo

Tabla 13 – Ubicación elementos Alumnos Grupo

- Seleccionar Sesión de Prácticas: esta actividad contiene una lista cuyos elementos son solo texto, por lo que al igual que en el caso de la selección de grupo de prácticas, se requiere únicamente de un descriptor personalizado para recuperar el elemento seleccionado correspondiente; pero no para definirlo.

Layout	<ul style="list-style-type: none"> <li>- activity_sesiones</li> <li>- opciones_lista_layout: descriptor para contener un elemento de texto. Permite recuperar la opción seleccionada entre todas las presentadas por cada sesión de prácticas.</li> </ul>
GUI	- SesionesActivity
BO	- SesionPracticasAdapter: adaptador para manejar las opciones desplegadas por cada elemento de la lista de sesiones.
Menu	- menu_sesiones

Tabla 14 – Ubicación elementos Seleccionar Sesión

- Crear Sesión de Prácticas: esta actividad es un formulario simple, cuyos objetos tienen correspondencia directa.

Layout	- activity_crear_sesion
GUI	- CrearSesionActivity
BO	
Menu	- menu_nueva_sesion

Tabla 15 – Ubicación elementos Crear Sesión

- Alumnos de la Sesión de Prácticas: esta actividad implementa una lista que no es solo texto, por lo que se necesita un descriptor de diseño que se mostrará por cada elemento individual de la lista.

Layout	- activity_alumnos_sesion: descriptor de diseño principal de la actividad. Contiene básicamente una lista en la que se mostrarán los datos. - elemento_lista_sesion: contiene el diseño de cada elemento de la lista. - grupo_list_layout: Se ha reutilizado este layout de la actividad Alumnos Grupo Prácticas. Permite recuperar el checkbox seleccionado de un elemento_lista_sesion.
GUI	- AlumnosSesionActivity: implementa la lógica de activity_alumnos_sesion.
BO	- AlumnoSesionListAdapter: contiene el manejador de cada elemento de la lista elemento_lista_sesion.
Menu	- menu_alumnos_grupo: se ha reutilizado el mismo diseño de menú para esta actividad que en Alumnos Grupo Prácticas.

Tabla 16 – Ubicación elementos Alumnos Sesión

- Ayuda: no es una actividad en sí misma, ya que solamente sirve como visor de información de ayuda y será instanciado por cada actividad cuando se pulse la opción correspondiente de su menú. Únicamente necesita de un descriptor con un control webView, en el que se mostrará el recurso de la carpeta assets que corresponda a la actividad que se está ejecutando actualmente. Cada actividad será la encargada de cargar el recurso correcto.

Layout	- ayuda
GUI	
BO	
Menu	

Tabla 17 – Ubicación elementos Ayuda

- AcercaDe: no es una actividad en sí misma, ya que solamente sirve como visor de información general de la aplicación y será instanciado por cada actividad login cuando se pulse la opción correspondiente de su menú. Está formada por un conjunto de controles estáticos con los datos preestablecidos.

Layout	- about
GUI	
BO	
Menu	

*Tabla 18 – Ubicación elementos AcercaDe*

## 1.6 Bibliografía

- [1] Servicio de Informática. (2016). Servicios Web de la Universidad de Jaén. 10/05/2016, de Universidad de Jaén Sitio web: <http://dv.ujaen.es/docencia/webservice/soap/server.php>
- [2] Servicio de Informática. (2016). Descriptor WSDL de los Servicios Web de la Universidad de Jaén. 10/05/2016, de Universidad de Jaén Sitio web: <http://dv.ujaen.es/docencia/webservice/soap/server.php?wsdl>
- [3] Equipo Desarrollo KSoap-android. (2015). Referencia KSoap-android. 15/07/2015, de Github Sitio web: <http://simpligility.github.io/ksoap2-android/howto.html>
- [4] Equipo Desarrollo Android. (2010). Referencia NFC. 15/07/2015, de Google Sitio web: <https://developer.android.com/reference/android/nfc/package-summary.html>
- [5] Equipo Desarrollo Android. (2010). Referencia SQLite. 16/07/2015, de Google Sitio web: <https://developer.android.com/reference/android/database/sqlite/SQLiteDatabases.html?hl=es>
- [6] James Revelo. (2014). Tutorial De Bases De Datos SQLite En Aplicaciones Android. 22/02/2016, de Hermosa Programación Sitio web: <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>
- [7] Equipo Desarrollo Android. (2010). Referencia API Android. 16/07/2015, de Google Sitio web: <http://developer.android.com/intl/es/guide/index.html>
- [8] José Manuel Jiménez Bravo. (2014). Aplicación para el control de acceso a dependencias y edificios basado en NFC (Near Field Communication) para terminales Android. 14/07/2015, de Universidad de Jaén.

## 2 PLANOS

## 3 PLIEGO DE CONDICIONES

### 3.1 Licencias del software

AsistenciaUJA se publica bajo licencia Creative Commons (en adelante, CC) del tipo “Reconocimiento -no comercial- sin obra derivada”, de forma que los usuarios estarán obligados a citar y reconocer los créditos de los trabajos de forma visible mediante mención directa al trabajo y autor, a la URL donde está alojada la aplicación, indicando que ésta se rige por licencia CC y manteniendo intactos los principios en los que se basan las demás licencias utilizadas.

El proveedor de servicios de la plataforma de docencia virtual de la Universidad de Jaén es Ilias: un sistema de gestión de enseñanza de código abierto, publicado bajo licencia GPL (del inglés GNU General Public License), la cual garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software, junto con la publicación del código fuente en un formato legible.

Android es el sistema operativo móvil para el que está diseñado la aplicación AsistenciaUJA. Dicho sistema operativo se publica bajo licencia Apache 2.0 y GPL v2, garantizando libertad de estudio, copia y modificación del software; pero requiriendo la conservación del aviso de derecho de autor y de descargo de responsabilidad en su caso; sin necesidad de publicar el código fuente en caso de modificación y sin posibilidad de aplicar restricciones en la distribución GPL de software modificado que restrinja la libertad del usuario final.

Android Studio es el entorno utilizado para desarrollar la aplicación, el cual está publicado bajo licencia Apache 2.0, la cual permite libertad de uso, estudio, copia y modificación del software; aunque no obliga a la publicación del código fuente en caso de modificación.

Ksoap es la librería utilizada para soportar comunicaciones a través del protocolo SOAP con los servicios web de la Universidad de Jaén. Dicha librería está publicada bajo licencia MIT, la cual permite libertad de estudio, copia y modificación del software; pero requiriendo la conservación del aviso de descargo de responsabilidad en su caso.

### 3.2 Características del hardware

#### 3.2.1 Terminal móvil

Para poder utilizar la aplicación AsistenciaUJA correctamente, se necesita un terminal móvil con las siguientes características:

- Android 2.3 como mínimo, aunque se recomienda 5.0 o posterior.
- Chip NFC integrado.
- Habilitado para comunicaciones de datos inalámbricas.
- Acceso a Internet.
- 5 MB de espacio libre.

### 3.2.2 Tarjetas NFC

Las tarjetas NFC de los alumnos deberán estar en línea con la especificación Type

2 TAG del NFC Forum:

- Basada en ISO/IEC 14443A
  - o Frecuencia de funcionamiento: 13.56MHz
  - o Velocidad de transferencia:106Kbps
  - o Sin batería
  - o Alcance hasta 100m
- Capaz de lectura y escritura.
- Memoria hasta 2 KB.

### 3.3 Requisitos de los usuarios

El tutor debe cumplir las siguientes características:

- Tener cuenta activa en la plataforma de docencia virtual de la Universidad de Jaén.
- Disponer de permisos de administrador para sus asignaturas en la plataforma de docencia virtual.
- Crear la carpeta asistencia en cada asignatura a monitorizar.
- Crear el fichero alumnos.csv con la información de los alumnos, dentro de la carpeta asistencia de cada asignatura.

## 4 PRESUPUESTO

A continuación, se presenta un desglose de las tareas necesarias para la realización de este proyecto y su coste asociado:

### 4.1 Diseño del Proyecto

Descripción	Cantidad	P. Ud. (€)	TOTAL
Horas de desarrollo de las especificaciones funcionales	15,0	17,00 €	255,00 €
Horas de desarrollo de la arquitectura del sistema	15,0	17,00 €	255,00 €
Horas de desarrollo de las especificaciones de diseño preliminares	8,0	17,00 €	136,00 €
Horas de desarrollo de las especificaciones de diseño detalladas	7,0	17,00 €	119,00 €
<b>Subtotal</b>	<b>45,0</b>	<b>17,00 €</b>	<b>765,00 €</b>

Tabla 19 - Presupuesto Partida Diseño

### 4.2 Desarrollo del Proyecto

Descripción	Cantidad	P. Ud. (€)	TOTAL
Desarrollar componentes	60,0	17,00 €	1.020,00 €
Obtener software	10,0	17,00 €	170,00 €
Obtener hardware	10,0	17,00 €	170,00 €
Desarrollar paquete de prueba de aceptación	3,0	17,00 €	51,00 €
Realizar prueba de unidad/integración	2,0	17,00 €	34,00 €
<b>Subtotal</b>	<b>85,0</b>	<b>17,00 €</b>	<b>1.445,00 €</b>

Tabla 20 - Presupuesto Partida Desarrollo

### 4.3 Entrega del Proyecto

Descripción	Cantidad	P. Ud. (€)	TOTAL
Instalar el sistema	1,0	17,00 €	17,00 €
Entrenar a los clientes	1,0	17,00 €	17,00 €
Realizar prueba de aceptación	1,0	17,00 €	17,00 €
Archivar material	1,0	17,00 €	17,00 €
<b>Subtotal</b>	<b>4,0</b>	<b>68,00 €</b>	<b>68,00 €</b>

Tabla 21 - Presupuesto Partida Entrega

### 4.4 Administración del proyecto

Descripción	Cantidad	P. Ud. (€)	TOTAL
-------------	----------	------------	-------

Reuniones/informes de progreso con el cliente	3,0	17,00 €	51,00 €
Administración de la configuración	2,0	17,00 €	34,00 €
Control de calidad	1,0	17,00 €	17,00 €
Administración global del proyecto	1,0	17,00 €	17,00 €
<b>Subtotal</b>	<b>7,0</b>	<b>68,00 €</b>	<b>119,00 €</b>

*Tabla 22 - Presupuesto Partida Administración*

#### 4.5 Resumen

DISEÑO DEL PROYECTO			<b>765,00 €</b>
DESARROLLO DEL PROYECTO			<b>1.445,00 €</b>
ENTREGA DEL PROYECTO			<b>68,00 €</b>
ADMINISTRACIÓN DEL PROYECTO			<b>119,00 €</b>
<b>Subtotales</b>	<b>141,0</b>	<b>170,00 €</b>	<b>2.397,00 €</b>
IVA	0,0	21%	<b>503,37 €</b>
<b>Total</b>	<b>141,0</b>	<b>170,21 €</b>	<b>2.900,37 €</b>

*Tabla 23 - Presupuesto Resumen*

El presupuesto total para este proyecto asciende a **dos mil novecientos euros con treinta y siete céntimos (2.900,37 €)**.