



ugr

Universidad
de **Granada**

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

E-DUCA+

PLATAFORMA DE APOYO A LA DOCENCIA

Autor

Ignacio Erenas Rodríguez

Director

Luis Castillo Vidal



Escuela Técnica Superior De Ingenierías Informática y de
Telecomunicación.

Granada, Julio de 2015

Índice

1. Resumen	2
2. Abstract.....	2
3. Motivación e Introducción.....	4
4. Objetivos del trabajo.....	6
5. Resolución del trabajo	7
5.2 Análisis y diseño.....	7
5.2.1 Metodología de desarrollo	7
5.2.2 Diagramas	36
5.2.3 Interfaces de Usuario	45
5.3 Implementación	54
5.3.1 Android.....	54
5.3.2 Servidor	55
6. Conclusiones y vías futuras	56
6.1 Académica	56
6.2 Profesional	56
7. Bibliografía.....	57

1. RESUMEN

E-duca+: Plataforma de apoyo a la docencia

Ignacio Erenas Rodríguez

Palabras clave: Educación, Software, Android, API, Desarrollo Ágil.

Resumen

Se ha desarrollado una aplicación para dispositivos Android cuya función es ayudar a los profesores en sus gestiones diarias en la escuela, de esta forma podrá llevar en un mismo dispositivo las tutorías y tareas que tiene programadas, las calificaciones de cada examen realizado así como la posibilidad de ponerse en contacto de una forma directa y sencilla con toda la comunidad educativa (profesores, padres y alumnos) a través de mensajes, la obtención de datos se realiza a través de la base de datos almacenada en un servidor, los datos se obtienen a través de una API desarrollada que facilita la inserción o modificación de nuevos datos, así como la creación de nuevo software para cualquier tipo de aplicación, ya sea móvil, web o de escritorio.

2. ABSTRACT

E-duca+: Plataforma de apoyo a la docencia

Ignacio Erenas Rodríguez

Keywords: Education, Software, Android, API, Agile Development.

Abstract

An educational Android platform has been developed offering the following functions: to improve the communication between the members of the educational community and help teachers making their work easy through the developed software, allowing actions like tutorial with parents and a direct communication with the students and the parents within an integrated message service.

It is also included some advantages for parents and children, therefore the software permits having- in the same application- a calendar for homework, the grades in the different tasks and the possibility of making tutorial appointments with different teachers.

The platform is based on an API (Application Programming Interface) development which allows the data exploitation in any kind operative system or application wanted (desktop or web app), the development of this API proposes two important advantages: in the first place it allows a great flexibility when adapting the platform to any

institution and in the second place, the effort for the development of new software will be smaller having basic functions defined.

Such functions make the insertion and the obtaining of the stored data in a SQL server data base which will have the role of middleware between the data base and the produced software.

The creation of this API permits the development of any type of application, web, desktop or mobile.

The Android application provides all this academic information stored all in one device, moreover teachers will have the possibility of adding tasks or grades wherever they are, even by an Internet connection or not, so then, the user will be noticed of every change they do in the server at the moment of the connection.

Consequently, once we have studied the three essential parts that compound the Project, we can differentiate three levels inside the Project:

- Back-End, formed by the Apache server and the SQL data base server
- Middleware, formed by the API, an intermediary between the back-end and the front-end.
- Front-End, Android app, although thanks to the advantages that the API offers the Front-end can be compound by any type of web or desktop application.

The integration of the GCM (Google Cloud Messaging) service makes the use of the “Push” notifications posible, in order to communicate the user immediately at the moment of new changes and he will be always noticed.

Having developed all this application by means of the agile software development Scrum provides a minimum cost charge when a needed change in the application, even though, making easier the ability of connecting new modules to give the application a higher usefulness.

Therefore, it has been concluded that an obtention of an Android application whose base for extracting the data from the server has been, at the same time, the creation of an API which allows the exploitation of the data based located in the server and then, the reduction of time when developing a new software.

3. MOTIVACIÓN E INTRODUCCIÓN

La educación siempre ha sido una materia en constante evolución, se ha buscado los mejores métodos de enseñanza, pero en los últimos tiempos debido a la rápida evolución de las tecnologías de la información y la comunicación (TIC), la gran mayoría de estas evoluciones han tenido como protagonista un dispositivo informático.

Estas evoluciones han tenido en varias ocasiones como objetivo mejorar la impartición teórica que el profesorado ofrece en el día a día, ayudar a los docentes en la gestión de las asignaturas que imparten o mejorar la comunicación entre los distintos miembros que componen la comunidad educativa.

Por ello, para aumentar las posibilidades y medios de comunicación, la inclusión de Internet en nuestro día a día ha sido determinante para la creación de diferentes plataformas web de apoyo a la docencia como *Moodle*¹ o *Séneca*², que además de apoyar la comunicación ha servido para dar una educación de mayor calidad.

Este proyecto tiene como objetivo principal el mejorar la comunicación entre las partes implicadas en la educación de los alumnos, pero sin dejar de lado la importancia que requiere la gestión que el profesorado debe realizar de cada una de sus asignaturas.

Hoy en día es indudable la gran importancia que están adquiriendo los denominados teléfonos inteligentes (*Smartphone*), y por tanto debe verse como una oportunidad para que el ámbito de la educación saque provecho de todas las ventajas que estos nos brindan.

El hecho de que la sociedad española sea el país de la Comunidad Europea que más utiliza este tipo de dispositivos³, ofrece la posibilidad de introducir en el día a día de la educación estos dispositivos como una herramienta más en el ámbito de la educación, ya sea para una simple comunicación entre las partes implicadas o para gestionar de una forma más avanzada los aspectos relacionados con la educación como tutorías o tareas.

El *software* desarrollado, aprovechando esta gran ventaja, facilita enormemente la comunicación entre los miembros de la comunidad educativa, pero sobre todo centrándose en la comunicación padre-profesor y viceversa.

La escuela podrá informar de forma directa y sin ningún tipo de intermediarios al padre ante cualquier circunstancia de la cual deba tener conocimiento con un servicio de mensajería instantánea cuyo uso es muy sencillo e incluso aquellos usuarios menos familiarizados con las nuevas tecnologías podrán aprovecharse de sus ventajas.

También facilita al profesorado su gestión de las asignaturas, ya que el profesor tendrá la capacidad de crear tareas y calificarlas dentro de la propia aplicación, además de una agenda para cada asignatura en la cual se le muestra las tareas que hay programadas.

Por otra parte cuentan con un sistema de tutorías para facilitar las comunicaciones en persona con los padres, además este sistema es bidireccional, es decir, un profesor puede solicitar una tutoría y viceversa.

Para los padres representa una gran ventaja la posibilidad de ver a diario el rendimiento de sus hijos en las distintas asignaturas, ya que recibirán automáticamente un aviso a su móvil cuando sus hijos reciban una nueva tarea o calificación.

Otra gran ventaja es el hecho de no ser necesaria la conexión a internet para realizar cualquier acción y que la aplicación pueda ser utilizada en cualquier lugar y en cualquier *smartphone*, incluso aunque este no disponga conexión de conexión de datos a través de móvil, ya que la actualización de los cambios se realizará de manera automática cuando el usuario disponga de conexión a internet.

4. OBJETIVOS DEL TRABAJO

Como ha sido mencionado en el apartado anterior, el objetivo principal del desarrollo de este trabajo es mejorar la comunicación en el ámbito educativo entre las distintas partes que lo componen.

Para esta mejora se ha hecho uso de la importancia que están adquiriendo hoy en día los *smartphone* y los diversos beneficios que estos nos aportan. Por ello la base de este proyecto ha sido una aplicación Android.

En un principio la idea del proyecto era una especie de “Productor de Aplicaciones para Apoyo a la Docencia”, dicha idea consistía en realizar una base muy sólida para a partir de ella poder generar *software* de una manera más sencilla.

En primer lugar para tener una buena base, la realización de un servidor el cual contara con una *Application Programming Interface* (API), que sería básica para proporcionar la información necesaria a las aplicaciones ya sean web o de escritorio de las distintas escuelas (de ahora en adelante plataformas). Dado que puede haber plataformas con peculiaridades, se debe construir una API cuya modificación tuviera un impacto mínimo en tiempo de desarrollo y eficiencia.

Logrado: Una modelado de datos que permite almacenar la información necesaria para la gestión escolar.

Logrado: Una API que satisface las necesidades tanto de una aplicación móvil como de cualquier otro tipo de aplicación (escritorio, web, etc...) web, además de tener una gran facilidad para la creación de cambios en el código para una plataforma concreta.

En segundo lugar, el desarrollo de una aplicación Android, la cual permitiera una fácil personalización para generar una nueva aplicación en un breve período de tiempo.

Logrado: Una aplicación Android totalmente estable, con funcionalidad online y offline, abstrayendo los estilos de los diferentes componentes en simples variables para así poder conseguir una personalización rápida y eficaz.

5. RESOLUCIÓN DEL TRABAJO

5.2 Análisis y diseño

5.2.1 Metodología de desarrollo

Para la realización del proyecto se ha optado por la metodología de desarrollo ágil *Scrum*, el cual es iterativo e incremental, no se basa en el seguimiento estricto de un plan, si no en la adaptación continua a las circunstancias del proyecto, cada iteración del proyecto es llamado sprint y a la finalización de cada uno se crea un incremento de software utilizable.

La facilidad de *Scrum* para adaptarse al cambio era muy importante, ya que al haber un único desarrollador, no había múltiples puntos de vista, y en caso de realizar alguna modificación, esta se podía llevar a cabo en un tiempo razonable.

Para comprender mejor esta metodología de desarrollo, procedo a hacer una breve explicación de los elementos que participan en *Scrum*.

Hay tres grande grupos en *Scrum*: roles, reuniones y artefactos, cada uno contiene sus propios elementos que procedo a describir:

- Roles: diferentes roles que tienen los actores que forman esta metodología de desarrollo, que son *product owner*, *scrum master*, *development team*.
 - *Product Owner*: Es el representante del cliente e intermediario de éste y el equipo de desarrollo.
 - *Scrum Master*: Es el director del equipo, el cual gestiona el proceso y garantiza las metodologías que se emplean.
 - *Development Team*: Conjunto de desarrolladores que desempeñan el trabajo de cada sprint.

- Reuniones: las reuniones que se tienen durante la realización del proyecto son *sprint planning*, *daily meeting*, *sprint review* y *sprint retrospective*.
 - *Sprint Planning*: Reunión previa al comienzo de cada sprint para identificar el objetivo principal del mismo y las tareas que se desarrollarán.
 - *Daily Meeting*: Reunión diaria para hacer un breve repaso del sprint y las actividades realizadas por cada desarrollador.
 - *Sprint Review*: Sirve para obtener una recogida de información sobre el estado del proyecto, analizar si se necesitan mejoras en el trabajo realizado o añadir elementos al *product backlog*.
 - *Sprint Retrospective*: Reunión del equipo de desarrollo para analizar y revisar lo realizado hasta el momento y organizar el trabajo para ser más eficiente.
- Artefactos: está compuesto por *product backlog*, *sprint backlog* y *burndown charts*
 - *Product Backlog*: es una lista ordenada y priorizada de las historias de usuario que podrían ser necesarias en el proyecto.
 - *Sprint Backlog*: selección de elementos del *product backlog* para el sprint actual, es decir, aquellos conjuntos de necesidades a implementar en el sprint actual.
 - *Burndown Charts*: gráfica representativa del trabajo pendiente para el equipo de desarrollo, puede ser relacionada con un sprint concreto o con el proyecto.

Tras varias reuniones de trabajo tanto con el director del proyecto como con usuarios potenciales de esta aplicación, se ha obtenido el siguiente listado de historias de usuario que conformarán nuestro *product backlog*.

Identificador: HU.1	Un usuario puede autenticarse como profesor, padre o alumno.	
Descripción: Un usuario puede iniciar sesión sin importar el rol al cual este pertenezca.		
	Prioridad: 1	Iteración: 1
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en el servidor.• Construir la interfaz de usuario (IU) para iniciar sesión.• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Inicio de sesión correcto como profesor.• Inicio de sesión correcto como padre.• Inicio de sesión correcto como alumno.		
Observaciones:		

Identificador: HU.2	Un usuario obtendrá todos sus usuarios relacionados cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática los usuarios relacionados con él.		
	Prioridad: 1	Iteración: 1
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en Android.• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Usuarios obtenidos de forma correcta iniciando sesión como profesor.• Usuarios obtenidos de forma correcta iniciando sesión como padre.• Usuarios obtenidos de forma correcta iniciando sesión como alumno.		
Observaciones: Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.		

Identificador: HU.3	Un usuario puede enviar mensajes sin tener conexión a Internet.	
Descripción: Un usuario puede enviar un mensaje a otro usuario, y en caso de no tener conexión en el momento de mandar el mensaje, mandarlo de forma automática al tener conexión.		
	Prioridad: 1	Iteración: 1
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en Android.• Construir la interfaz de usuario (IU) para enviar mensajes.• Construir la capa de datos en el servidor.• Implementar conexión con el servidor para enviar mensajes.• Implementar función de inserción de mensajes en el servidor.• Implementar envío automático de mensajes cuando el dispositivo se conecta a una red.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario envía un mensaje al servidor y se almacena de forma correcta en la base de datos del dispositivo Android.• Un usuario envía un mensaje con conexión y se almacena de forma correcta en la base de datos del servidor.• Un usuario envía un mensaje sin conexión y se comprueba el envío automático al conectarse a una red.		
Observaciones:		

Identificador: HU.4	Un usuario puede buscar a otro usuario para iniciar una conversación.	
Descripción: A un usuario se le mostrará una lista de usuarios que coinciden con el nombre y apellidos que busca mientras escribe.		
	Prioridad: 1	Iteración: 2
Tareas relacionadas: <ul style="list-style-type: none">• Implementar funcionalidad de búsqueda en Android.• Construir la interfaz de usuario (IU).		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario buscará distintos usuarios y se comprobará su correcta aparición en la lista.		
Observaciones: Depende de la HU.2 y la correcta obtención de los usuarios al iniciar sesión.		

Identificador: HU.5	Un usuario puede borrar mensajes sin tener conexión a Internet.	
Descripción: Un usuario puede eliminar un mensaje, y en caso de no tener conexión en el momento de borrar el mensaje, se eliminará de la interfaz gráfica y se deberá enviar la modificación al servidor de forma automática al tener conexión.		
	Prioridad: 2	Iteración: 9
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario para no mostrar el mensaje eliminado.• Modificar interfaz de usuario mostrar la opción de borrar mensaje.• Implementar funcionalidad de borrado en la base de datos en Android.• Implementar funcionalidad en el servidor para marcar el mensaje como no visible al usuario.• Implementar conexión con el servidor para modificar el mensaje.• Implementar conexión automática con el servidor para notificar la modificación en el mensaje.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario elimina un mensaje y se modifica de forma correcta en la base de datos del dispositivo Android.• Un usuario elimina un mensaje y se modifica la IU para no mostrar el mensaje.• Un usuario elimina un mensaje con conexión y se modifica de forma correcta en la base de datos del servidor.• Un usuario elimina un mensaje sin conexión y se comprueba la modificación automática en la base de datos del servidor al conectarse a una red.		
Observaciones:		

Identificador: HU.6	Un usuario puede borrar conversaciones sin tener conexión a internet.	
Descripción: Un usuario puede eliminar una conversación escogida, y en caso de no tener conexión en el momento de borrar la conversación, se eliminará de la interfaz gráfica y se deberá enviar la modificación al servidor de forma automática al tener conexión.		
	Prioridad: 2	Iteración: 9
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario para no mostrar la conversación relacionada.• Modificar interfaz de usuario mostrar la opción de borrar conversación.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario elimina una conversación y se modifica de forma correcta los mensajes de ésta en la base de datos del dispositivo Android.• Un usuario elimina una conversación y se modifica la IU para no mostrar la conversación.		
Observaciones: Depende de la HU.5, ya que se realiza un uso de esta historia de uso reiterado sobre cada mensaje de la conversación.		

Identificador: HU.7	Un usuario puede marcar como leído un mensaje.	
Descripción: Un usuario puede marcar como leído un mensaje no leído.		
	Prioridad: 2	Iteración: 9
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario para mostrar el mensaje como leído.• Modificar interfaz de usuario mostrar la opción de marcar mensaje leído.• Implementar funcionalidad de marcar como leído en la base de datos en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario marca como leído un mensaje y se modifica de forma correcta en la base de datos del dispositivo Android.• Un usuario marca como leído un mensaje y se modifica la IU para mostrar el mensaje leído.		
Observaciones:		

Identificador: HU.8	Un usuario puede marcar como leída una conversación.	
Descripción: Un usuario puede marcar como leídos los mensajes de una conversación.		
	Prioridad: 2	Iteración: 9
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario mostrar la opción de marcar conversación leído.• Implementar funcionalidad de marcar como leída una conversación en la base de datos en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario marca como leído una conversación y se modifica de forma correcta los mensajes de esta en la base de datos del dispositivo Android.		
Observaciones:		

Identificador: HU.9	Un usuario deberá ser notificado al móvil cuando reciba un mensaje nuevo.	
Descripción: Cuando un usuario reciba un mensaje, se mostrará una notificación aunque no se esté utilizando la aplicación.		
	Prioridad: 1	Iteración: 2
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en el servidor para la gestión de dispositivos registrados.• Construir la IU de las notificaciones.• Implementar funcionalidad de las notificaciones en Android.• Implementar funcionalidad de registro del dispositivo para la recepción de notificaciones.• Implementar función para envío de notificaciones en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario recibirá una notificación y se comprobará la correcta inserción del mensaje en la base de datos en Android.• Un usuario pulsará sobre la notificación y se iniciará la aplicación en la pestaña de mensajes.		
Observaciones:		

Identificador: HU.10	Un usuario obtendrá todos sus mensajes cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática los mensajes.		
	Prioridad: 1	Iteración: 3
Tareas relacionadas: <ul style="list-style-type: none">• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Mensajes enviados y recibidos obtenidos de forma correcta y almacenados en la base de datos de Android• No obtener aquellos mensajes borrados con anterioridad.		
Observaciones: Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.		

Identificador: HU.11	Un profesor o padre puede crear nuevas tutorías sin tener conexión a Internet.	
Descripción: Un usuario con rol de profesor o padre puede crear nuevas tutorías, se actualizará la interfaz gráfica, y en caso de no tener conexión en el momento de crear la tutoría, mandarla de forma automática al tener conexión.		
	Prioridad: 1	Iteración: 3
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para crear tutorías.• Construir la capa de datos en Android.• Construir la capa de datos en el servidor.• Implementar conexión con el servidor para crear tutorías.• Implementar función de inserción de tutorías en el servidor.• Implementar envío automático de tutorías cuando el dispositivo se conecta a una red.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de alumno no puede crear nuevas tutorías.• Un profesor crea la tutoría en base a un alumno existente y el padre solicita la tutoría a un profesor existente de un hijo existente.• Un profesor o padre envía una tutoría al servidor y se almacena de forma correcta en la base de datos del dispositivo Android.• Un profesor o padre envía una tutoría con conexión y se almacena de forma correcta en la base de datos del servidor.• Un profesor o padre envía una tutoría sin conexión y se comprueba el envío automático al conectarse a una red.		
Observaciones: Depende de la HU.2 y la correcta obtención de los usuarios al iniciar sesión para solicitar las tutorías a los usuarios asociados.		

Identificador: HU.12	Un profesor o padre puede modificar tutorías existentes sin tener conexión a Internet.	
Descripción: Un usuario con rol de profesor o padre puede modificar nuevas tutorías, se actualizará la interfaz de usuario, y en caso de no tener conexión en el momento de modificar la conexión, enviar la modificación de forma automática al tener conexión.		
	Prioridad: 2	Iteración: 10
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para modificar tutorías.• Modificar interfaz de usuario para mostrar la opción de editar tutorías.• Implementar conexión con el servidor para modificar tutorías.• Implementar función de modificación de tutorías en el servidor.• Implementar envío automático de tutorías cuando el dispositivo se conecta a una red.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de alumno no puede modificar una tutoría.• Un profesor o padre modifica una tutoría y se modifica de forma correcta en la base de datos del dispositivo Android.		

<ul style="list-style-type: none"> • Un profesor o padre modifica una tutoría con conexión y se almacena de forma correcta en la base de datos del servidor. • Un profesor o padre modifica una tutoría sin conexión y se comprueba el envío automático al conectarse a una red.
Observaciones:

Identificador: HU.13	Un usuario puede ver la ficha de una tutoría.	
Descripción: Un usuario puede ver una tutoría relacionada con él, se mostrarán en primer lugar agrupadas por estados y al seleccionar una tutoría de un estado se mostrará la ficha.		
	Prioridad: 1	Iteración: 4
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) de la ficha.• Construir la interfaz de usuario (IU) de agrupamiento de tutorías.• Implementar la funcionalidad para obtener las tutorías agrupadas por estado.• Implementar la funcionalidad para obtener una tutoría concreta.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario tendrá agrupadas las tutorías por estados.• Un usuario podrá acceder a la ficha de una tutoría concreta a partir del anterior agrupamiento.		
Observaciones:		

Identificador: HU.14	Un usuario puede buscar a otro usuario para iniciar una conversación.	
Descripción: A un usuario con rol de profesor o padre le saldrá una ventana emergente con las opciones de cambio de estado en la tutoría.		
	Prioridad: 3	Iteración: 11
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario para mostrar la ventana con las opciones.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de alumno no puede modificar el estado de una tutoría.		
Observaciones: Depende de la HU.12, ya que la funcionalidad de modificación de tutoría es reutilizada.		

Identificador: HU.15	Un usuario deberá ser notificado al móvil cuando reciba una nueva tutoría o alguna modificación sobre estas.	
Descripción: Cuando un usuario reciba una tutoría o alguna modificación sobre alguna existente, se mostrará una notificación aunque no se esté utilizando la aplicación.		
	Prioridad: 1	Iteración: 4
Tareas relacionadas: <ul style="list-style-type: none">• Construir la IU de las notificaciones.• Implementar funcionalidad de las notificaciones en Android.• Implementar función para envío de notificaciones en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación:		

- Un usuario recibirá una notificación y se comprobará la correcta inserción o modificación de la tutoría en la base de datos en Android.
- Un usuario pulsará sobre la notificación y se iniciará la aplicación en la pestaña de tutorías.

Observaciones:

La capa de datos de los dispositivos registrados y la implementación de dicho registro ya está realizada en la HU.9, por tanto no deben ser implementadas en este caso

Identificador: HU.16	Un usuario obtendrá todas sus tutorías cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática las tutorías.		
	Prioridad: 1	Iteración: 4
Tareas relacionadas: <ul style="list-style-type: none">● Implementar conexión con el servidor.● Implementar función en el servidor.● Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">● Tutorías relacionadas con el usuario obtenidas de forma correcta y almacenadas en la base de datos de Android		
Observaciones: Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.		

Identificador: HU.17	Un profesor puede asignar nuevas tareas a asignaturas sin tener conexión a Internet.	
Descripción: Un usuario con rol de profesor puede crear nuevas tareas, se actualizará la interfaz gráfica, y en caso de no tener conexión en el momento de crear la tarea, mandarla de forma automática al tener conexión.		
	Prioridad: 1	Iteración: 5
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para crear tareas.• Construir la capa de datos en Android.• Construir la capa de datos en el servidor.• Implementar conexión con el servidor para crear tareas.• Implementar función de inserción de tareas en el servidor.• Implementar envío automático de tareas cuando el dispositivo se conecta a una red.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de padre o alumno no puede crear nuevas tareas.• Un profesor crea la tarea asignándola a una asignatura existente.• Un profesor crea la tarea y se almacena de forma correcta en la base de datos del dispositivo Android.• Un profesor envía una tarea con conexión y se almacena de forma correcta en la base de datos del servidor.		

- Un profesor envía una tarea sin conexión y se comprueba el envío automático al conectarse a una red.

Observaciones:

Depende de la HU.20 y la correcta obtención de las asignaturas al iniciar sesión, necesario para asignar las tareas.

Identificador: HU.18	Un profesor puede modificar tareas existentes sin tener conexión a Internet.	
Descripción: Un usuario con rol de profesor puede modificar tareas existentes, se actualizará la interfaz gráfica, y en caso de no tener conexión en el momento de crear la tarea, mandarla de forma automática al tener conexión.		
	Prioridad: 2	Iteración: 10
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para modificar tareas.• Modificar interfaz de usuario para mostrar la opción de editar tareas.• Implementar conexión con el servidor para modificar tareas.• Implementar función de modificación de tareas en el servidor.• Implementar envío automático de tareas cuando el dispositivo se conecta a una red.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de alumno o padre no puede modificar una tarea.• Un profesor modifica una tarea y se modifica de forma correcta en la base de datos del dispositivo Android.• Un profesor modifica una tarea con conexión y se almacena de forma correcta en la base de datos del servidor.• Un profesor modifica una tarea sin conexión y se comprueba el envío automático al conectarse a una red.		
Observaciones:		

Identificador: HU.19	Un usuario puede ver la ficha de una tarea existente sin tener conexión a Internet.	
Descripción: Un usuario puede ver una ficha de una tarea relacionada con él, se mostrarán en primer lugar agrupadas por asignaturas y al seleccionar una tarea de una asignatura se mostrará la ficha.		
	Prioridad: 1	Iteración: 5
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) de la ficha.• Construir la interfaz de usuario (IU) de agrupamiento de tareas.• Implementar la funcionalidad para obtener las tareas agrupadas por asignaturas.• Implementar la funcionalidad para obtener una tarea concreta.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario tendrá agrupadas las tareas por asignaturas.• Un usuario podrá acceder a la ficha de una tarea concreta a partir del anterior agrupamiento.		
Observaciones:		

Identificador: HU.20	Un usuario obtendrá todas sus tareas relacionadas cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática las tareas.		
	Prioridad: 1	Iteración: 5
Tareas relacionadas: <ul style="list-style-type: none">• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Tareas relacionadas con el usuario obtenidas de forma correcta y almacenadas en la base de datos de Android		
Observaciones: Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.		

Identificador: HU.21	Un usuario deberá ser notificado al móvil cuando reciba una nueva tarea o alguna modificación sobre estas.	
Descripción: Cuando un usuario reciba una tarea o alguna modificación sobre alguna existente, se mostrará una notificación aunque no se esté utilizando la aplicación.		
	Prioridad: 1	Iteración: 6
Tareas relacionadas: <ul style="list-style-type: none">• Construir la IU de las notificaciones.• Implementar funcionalidad de las notificaciones en Android.• Implementar función para envío de notificaciones en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario recibirá una notificación y se comprobará la correcta inserción o modificación de la tarea en la base de datos en Android.• Un usuario pulsará sobre la notificación y se iniciará la aplicación en la pestaña de tareas.		
Observaciones: La capa de datos de los dispositivos registrados y la implementación de dicho registro ya está realizada en la HU.9, por tanto no deben ser implementadas en este caso		

Identificador: HU.22	Un usuario obtendrá todas sus asignaturas relacionadas cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática las asignaturas.		
	Prioridad: 1	Entrega: 6
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en Android.• Construir la capa de datos en el servidor.• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Asignaturas relacionadas con el usuario obtenidas de forma correcta y almacenadas en la base de datos de Android		
Observaciones: Depende de la HU.2 y la correcta obtención de los usuarios al iniciar sesión.		
Identificador: HU.23	Un usuario puede ver una ficha de las asignaturas relacionadas con el sin tener conexión a Internet.	
Descripción: Un usuario puede ver una ficha de una asignatura relacionada con él.		
	Prioridad: 1	Iteración: 8
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) de la ficha.• Implementar la funcionalidad para obtener las asignaturas relacionadas con un usuario.• Implementar la funcionalidad para obtener una tarea concreta.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario podrá acceder a la ficha de una tutoría concreta a partir del anterior agrupamiento.		
Observaciones:		

Identificador: HU.24	Un usuario puede ver un calendario con las tareas de una asignatura sin tener conexión a Internet.	
Descripción: En la ficha de las asignaturas se añadirá un calendario, el cual marcará las fechas en las cual haya una tarea asignada a dicha asignatura.		
	Prioridad: 3	Iteración: 12
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) del calendario.		

<ul style="list-style-type: none"> • Implementar funcionalidad del calendario para poder crear tareas a partir de una fecha seleccionada. • Implementar funcionalidad del calendario para mostrar las tareas marcadas en el calendario.
Pruebas de aceptación: <ul style="list-style-type: none"> • Un usuario con rol de padre o alumno no podrá crear una tarea a través del calendario. • La creación de una tarea a partir de una fecha funciona de forma satisfactoria.
Observaciones: Relacionada con la HU.23, ya que se modifica su interfaz de usuario.

Identificador: HU.25	Un profesor puede calificar la tarea de un alumno.	
Descripción: Un usuario con rol de profesor puede calificar tareas existentes y se actualizará la interfaz gráfica.		
	Prioridad: 1	Iteración: 7
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para calificar tareas.• Construir la capa de datos en Android.• Construir la capa de datos en el servidor.• Implementar conexión con el servidor para calificar tareas.• Implementar función de inserción de calificación de tareas en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de padre o alumno no puede calificar tareas.• Un profesor califica una tarea ya existente.• Un profesor califica la tarea y se almacena de forma correcta en la base de datos del dispositivo Android.• Un profesor califica una tarea con conexión y se almacena de forma correcta en la base de datos del servidor.		
Observaciones: Depende de la HU.22 y la correcta obtención de las tareas al iniciar sesión, necesario para calificar las tareas.		

Identificador: HU.26	Un profesor puede modificar la calificación de un alumno.	
Descripción: Un usuario con rol de profesor puede modificar calificaciones existentes y se actualizará la interfaz gráfica.		
	Prioridad: 2	Iteración: 10
Tareas relacionadas: <ul style="list-style-type: none">• Implementar conexión con el servidor para modificar tareas.• Implementar función de modificación de tareas en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación:		

<ul style="list-style-type: none"> • Un usuario con rol de padre o alumno no puede modificar calificaciones. • Un profesor modifica una calificación y se almacena de forma correcta en la base de datos del dispositivo Android. • Un profesor modifica una calificación y se almacena de forma correcta en la base de datos del servidor.
Observaciones:

Identificador: HU.27	Un padre o alumno puede ver la calificación asociada a una tarea en su ficha.	
Descripción: Modificación de la ficha de una tarea para los usuarios con rol de padre y alumno, para mostrar la calificación asociada a dicha tarea.		
	Prioridad: 1	Iteración: 8
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario (IU) para mostrar la calificación.• Implementar función en Android para mostrar un listado de tareas calificadas.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario con rol de profesor la ficha de tarea siempre es general, no muestra ningún campo relacionado con la calificación.• No mostrar ningún campo relacionado con la calificación en la ficha de la tarea, si esta aún no ha sido calificada.		
Observaciones: Relacionada con la HU.19, ya que se modifica su interfaz de usuario.		

Identificador: HU.28	Un usuario obtendrá todas sus calificaciones relacionadas cuando inicie sesión.	
Descripción: Un usuario al iniciar sesión deberá descargarse de forma automática las calificaciones.		
	Prioridad: 1	Iteración: 8
Tareas relacionadas: <ul style="list-style-type: none">• Implementar conexión con el servidor.• Implementar función en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Calificaciones relacionadas con el usuario obtenidas de forma correcta y almacenadas en la base de datos de Android		
Observaciones:		

Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.

Identificador: HU.29	Un usuario deberá ser notificado al móvil cuando reciba una nueva calificación o alguna modificación sobre estas.	
Descripción: Cuando un usuario reciba una calificación o alguna modificación sobre alguna existente, se mostrará una notificación aunque no se esté utilizando la aplicación.		
	Prioridad: 1	Iteración: 7
Tareas relacionadas: <ul style="list-style-type: none">• Construir la IU de las notificaciones.• Implementar funcionalidad de las notificaciones en Android.• Implementar función para envío de notificaciones en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario recibirá una notificación y se comprobará la correcta inserción o modificación de la calificación en la base de datos en Android.• Un usuario pulsará sobre la notificación y se iniciará la aplicación en la pestaña de calificaciones.		
Observaciones:		

Identificador: HU.30	Un usuario puede cerrar sesión.	
Descripción: Un usuario puede cerrar sesión en el dispositivo.		
	Prioridad: 3	Iteración: 11
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario (IU) para mostrar la opción de cerrar sesión.• Implementar función en Android para cerrar sesión.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario cierra sesión y es enviado a la pantalla de inicio de sesión.• La base de datos de Android ha sido eliminada al cerrar sesión.		
Observaciones:		

Identificador: HU.31	Un usuario puede cambiar su contraseña.	
Descripción: Un usuario puede cambiar su contraseña de acceso desde el dispositivo.		
	Prioridad: 2	Iteración: 11
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario (IU) para mostrar la opción de cambiar contraseña.• Construir la interfaz de usuario (IU) del cambio de contraseña.• Implementar conexión con el servidor para cambiar la contraseña.• Implementar función de cambio de contraseña en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Se deberá informa al usuario del resultado de la operación (éxito o error) por pantalla.• Para cambiar la contraseña se debe solicitar la contraseña actual y validar que es correcta.• Para cambiar la contraseña se pedirá dos veces la nueva contraseña para evitar error al escribir la nueva.• Un usuario cambiará su contraseña y se comprobará la correcta modificación de la misma en la base de datos de Android y el servidor.		
Observaciones:		

Identificador: HU.32	Un usuario puede cambiar su imagen de perfil sin tener conexión a Internet.	
Descripción: Un usuario podrá cambiar su imagen de perfil desde el dispositivo Android, en caso de no tener conexión en el momento del cambio se deberá subir la nueva imagen automáticamente cuando el dispositivo tenga conexión.		
	Prioridad: 4	Iteración: 11
Tareas relacionadas: <ul style="list-style-type: none">• Construir la capa de datos en Android.• Construir la capa de datos en el servidor.• Modificar interfaz de usuario (IU) para mostrar la opción de cambiar imagen.• Construir la interfaz de usuario (IU) del cambio de imagen.• Implementar conexión con el servidor para cambiar la imagen.• Implementar función de cambio de imagen en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario debe poder seleccionar cualquier imagen almacenada en su móvil.• Un usuario debe poder recortar la imagen seleccionada para mostrar la parte que desee de la misma.• Un usuario cambia su imagen y se modifica de forma correcta en la base de datos del dispositivo Android.• Un usuario cambia su imagen con conexión y se almacena de forma correcta en la base de datos del servidor.• Un usuario cambia su imagen sin conexión y se comprueba el envío automático al conectarse a una red.		
Observaciones:		

Identificador: HU.33	Un usuario obtendrá las imágenes de los usuarios asociados.	
Descripción: Un usuario obtendrá las imágenes de los usuarios al iniciar sesión y comprobará de forma regular si ha habido algún cambio.		
	Prioridad: 4	Iteración: 12
Tareas relacionadas: <ul style="list-style-type: none">• Implementar función de envío de imagen en el servidor.• Gestionar respuesta del servidor en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Imágenes de los usuarios obtenidas de forma correcta y almacenadas en la base de datos de Android		
Observaciones: Es una ampliación de HU.1, esta historia de usuario se realizará al iniciar sesión.		

Identificador: HU.34	Un usuario puede ver su conversación con otro usuario.	
Descripción: Un usuario podrá cambiar su imagen de perfil desde el dispositivo Android, en caso de no tener conexión en el momento del cambio se deberá subir la nueva imagen automáticamente cuando el dispositivo tenga conexión.		
	Prioridad: 1	Iteración: 3
Tareas relacionadas: <ul style="list-style-type: none">• Construir la interfaz de usuario (IU) para mostrar la conversación.• Implementar la funcionalidad para obtener la conversación con el usuario.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario podrá enviar mensajes nuevos desde la misma pantalla de la conversación.• El último mensaje de cada conversación aparecerá en un buzón de entrada, a partir del cual se accede a la conversación.		
Observaciones:		

Identificador: HU.35	Un usuario puede marcar como leída una tutoría.	
Descripción: Un usuario puede marcar como leída una tutoría no leída.		
	Prioridad: 2	Iteración: 10
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario para mostrar la tutoría como leída.• Modificar interfaz de usuario mostrar la opción de marcar tutoría leída.• Implementar funcionalidad de marcar como leída en la base de datos en Android.		
Pruebas de aceptación: <ul style="list-style-type: none">• Un usuario marca como leída una tutoría y se modifica de forma correcta en la base de datos del dispositivo Android.• Un usuario marca como leída una tutoría y se modifica la IU para mostrar la tutoría leída.		
Observaciones:		

Identificador: HU.36	Un usuario puede marcar como leída una tutoría.	
Descripción: Un usuario puede marcar como leído un grupo de tutorías no leídas.		
	Prioridad: 2	Iteración: 10
Tareas relacionadas: <ul style="list-style-type: none">• Modificar interfaz de usuario mostrar la opción de marcar grupo de tutorías leídas.		

<ul style="list-style-type: none"> • Implementar funcionalidad de marcar como leído un grupo de tutorías en la base de datos en Android.
Pruebas de aceptación: <ul style="list-style-type: none"> • Un usuario marca como leído un grupo de tutorías y se modifica de forma correcta en la base de datos del dispositivo Android.
Observaciones:

Una vez tenemos todas las historias de usuario con sus tareas relacionadas, a continuación se muestra un listado de estas con su título estimación y prioridad.

La estimación es expresada en Puntos de Historia y la prioridad está medida en un rango de 1 a 3, siendo el valor 1 más prioritario.

Ident.	Título	Estim	Prio
HU.1	Un usuario puede autenticarse como profesor, padre o alumno.	½	1
HU.2	Un usuario obtendrá todos sus usuarios relacionados cuando inicie sesión.	2	1
HU.3	Un usuario puede enviar mensajes sin tener conexión a Internet.	3	1
HU.4	Un usuario puede buscar a otro usuario para iniciar una conversación.	2	1
HU.5	Un usuario puede borrar mensajes sin tener conexión a Internet.	2	2
HU.6	Un usuario puede borrar conversaciones sin tener conexión a internet.	1	2
HU.7	Un usuario puede marcar como leído un mensaje.	1	2
HU.8	Un usuario puede marcar como leída una conversación.	1	2
HU.9	Un usuario deberá ser notificado al móvil cuando reciba un mensaje nuevo.	3	1
HU.10	Un usuario obtendrá todos sus mensajes cuando inicie sesión.	2	1
HU.11	Un profesor o padre puede crear nuevas tutorías sin tener conexión a Internet.	2	1
HU.12	Un profesor o padre puede modificar tutorías existentes sin tener conexión a Internet.	1	2
HU.13	Un usuario puede ver la ficha de una tutoría.	½	1
HU.14	Un profesor o padre puede cambiar de estado una tutoría sin necesidad de entrar en la ficha de ésta.	1	3
HU.15	Un usuario deberá ser notificado al móvil cuando reciba una nueva tutoría o alguna modificación sobre estas.	3	1
HU.16	Un usuario obtendrá todas sus tutorías cuando inicie sesión.	2	1
HU.17	Un profesor puede asignar nuevas tareas a asignaturas sin tener conexión a Internet.	2	1
HU.18	Un profesor puede modificar tareas existentes sin tener conexión a Internet.	1	2

HU.19	Un usuario puede ver la ficha de una tarea existente sin tener conexión a Internet.	1	1
HU.20	Un usuario obtendrá todas sus tareas relacionadas cuando inicie sesión.	2	1
HU.21	Un usuario deberá ser notificado al móvil cuando reciba una nueva tarea o alguna modificación sobre estas.	3	1
HU.22	Un usuario obtendrá todas sus asignaturas relacionadas cuando inicie sesión.	2	1
HU.23	Un usuario puede ver una ficha de las asignaturas relacionadas con el sin tener conexión a Internet.	1	1
HU.24	Un usuario puede ver un calendario con las tareas de una asignatura sin tener conexión a Internet.	2	3
HU.25	Un profesor puede calificar la tarea de un alumno.	2	1
HU.26	Un profesor puede modificar la calificación de un alumno.	1	2
HU.27	Un padre o alumno puede ver la calificación asociada a una tarea en su ficha.	1	1
HU.28	Un usuario obtendrá todas sus calificaciones relacionadas cuando inicie sesión.	2	1
HU.29	Un usuario deberá ser notificado al móvil cuando reciba una nueva calificación o alguna modificación sobre estas.	3	1
HU.30	Un usuario puede cerrar sesión.	½	3
HU.31	Un usuario puede cambiar su contraseña.	½	2
HU.32	Un usuario puede cambiar su imagen de perfil sin tener conexión a Internet.	3	4
HU.33	Un usuario obtendrá las imágenes de los usuarios asociados.	3	4
HU.34	Un usuario puede ver su conversación con otro usuario.	1	1
HU.35	Un usuario puede marcar como leída una tutoría.	1	2
HU.36	Un usuario puede marcar como leído un grupo de tutorías.	1	2

Una vez tenemos las historias de usuario del proyecto, es necesario saber la velocidad estimada del equipo para realizar un plan de entregas, por tanto a continuación procedemos a calcular dicha velocidad y realizar el plan de entrega.

Tenemos un equipo de desarrollo formado por un programador, cuya dedicación al proyecto es de un 80%:

- La **duración de un sprint** en el proyecto será de 3 semanas.
- La estimación realizada para las historias de usuario (tareas del *product backlo*), son expresadas en días ideales de programación.
- Estimamos que un día ideal de programación se va a corresponder con 3-4 días de trabajo real.

Por tanto, tenemos:

$$1 \text{ iteración} = 3 \text{ semanas} = 15 \text{ días reales}$$

La **velocidad del equipo** de desarrollo medido en punto de historia es:

$$1 \text{ programador} * 15 = 15 \text{ días reales} \rightarrow \text{de 5 a 4 PH por sprint.}$$

Utilizaremos como velocidad estimada del equipo 5 puntos de historia

Esfuerzo total del proyecto = 58 PH

Velocidad del equipo = 5 PH

En base al esfuerzo necesario y la velocidad estimada, se realizarán 12 sprint durante el proyecto, la finalización de cada sprint conllevará media semana para realizar las reuniones siguientes: *sprint review* y *sprint planning* para planificar el siguiente sprint, a la finalización de la 6ª iteración, además de las dos reuniones se tendrá el *sprint restrospective*, lo cual añadirá una semana de retraso a la 7ª iteración.

A continuación mostraré las historias de usuario a realizar en cada iteración, las historias de usuario con una mayor prioridad serán desarrolladas en primer lugar.

La estimación de las sumas de las historias de usuario de cada iteración deben sumar 5 PH, pero en algunos casos debido a temas de desarrollo (dependencias de unas historias de usuario con otras), pueden variar levemente.

Sprint 1		Inicio: 15/09/2014
Ident.	Título de la historia de usuario	Estimación
HU. 1	Un usuario puede autenticarse como profesor, padre o alumno.	½
HU. 2	Un usuario obtendrá todos sus usuarios relacionados cuando inicie sesión.	2
HU. 3	Un usuario puede enviar mensajes sin tener conexión a Internet.	3

Sprint 2		Inicio: 8/10/2014
Ident.	Título de la historia de usuario	Estimación
HU. 9	Un usuario deberá ser notificado al móvil cuando reciba un mensaje nuevo.	3
HU. 4	Un usuario puede buscar a otro usuario para iniciar una conversación.	2

Sprint 3		Inicio: 3/11/2014
Ident.	Título de la historia de usuario	Estimación
HU. 34	Un usuario puede ver su conversación con otro usuario.	1
HU. 10	Un usuario obtendrá todos sus mensajes cuando inicie sesión.	2
HU. 11	Un profesor o padre puede crear nuevas tutorías sin tener conexión a Internet.	2

Sprint 4		Inicio: 26/11/2014
Ident.	Título de la historia de usuario	Estimación
HU. 13	Un usuario puede ver la ficha de una tutoría.	½
HU. 15	Un usuario deberá ser notificado al móvil cuando reciba una nueva tutoría o alguna modificación sobre estas.	3
HU. 16	Un usuario obtendrá todas sus tutorías cuando inicie sesión.	2

Sprint 5		Inicio: 22/12/2014
Ident.	Título de la historia de usuario	Estimación
HU. 17	Un profesor puede asignar nuevas tareas a asignaturas sin tener conexión a Internet.	1
HU. 19	Un usuario puede ver la ficha de una tarea existente sin tener conexión a Internet.	2
HU. 20	Un usuario obtendrá todas sus tareas relacionadas cuando inicie sesión.	2
Observaciones: Debido al período festivo se retrasa otra semana el inicio de este Sprint		

Sprint 6		Inicio: 12/1/2015
Ident.	Título de la historia de usuario	Estimación
HU. 21	Un usuario deberá ser notificado al móvil cuando reciba una nueva tarea o alguna modificación sobre estas.	3
HU. 22	Un usuario obtendrá todas sus asignaturas relacionadas cuando inicie sesión.	2

Sprint 7		Inicio: 9/2/2015
Ident.	Título de la historia de usuario	Estimación
HU. 25	Un profesor puede calificar la tarea de un alumno.	2

HU. 29	Un usuario deberá ser notificado al móvil cuando reciba una nueva calificación o alguna modificación sobre estas.	3
-----------	---	---

Sprint 8		Inicio: 4/3/2015
Ident.	Título de la historia de usuario	Estimación
HU. 23	Un usuario puede ver una ficha de las asignaturas relacionadas con el sin tener conexión a Internet.	1
HU. 27	Un padre o alumno puede ver la calificación asociada a una tarea en su ficha.	1
HU. 28	Un usuario obtendrá todas sus calificaciones relacionadas cuando inicie sesión.	2

Sprint 9		Inicio: 30/3/2015
Ident.	Título de la historia de usuario	Estimación
HU. 5	Un usuario puede borrar mensajes sin tener conexión a Internet.	2
HU. 6	Un usuario puede borrar conversaciones sin tener conexión a internet.	1
HU. 7	Un usuario puede marcar como leído un mensaje.	1
HU. 8	Un usuario puede marcar como leída una conversación.	1

Sprint 10		Inicio: 22/4/2015
Ident.	Título de la historia de usuario	Estimación
HU. 12	Un profesor o padre puede modificar tutorías existentes sin tener conexión a Internet.	1
HU. 18	Un profesor puede modificar tareas existentes sin tener conexión a Internet.	1

HU. 26	Un profesor puede modificar la calificación de un alumno.	1
HU. 35	Un usuario puede marcar como leída una tutoría.	1
HU. 36	Un usuario puede marcar como leído un grupo de tutorías.	1

Sprint 11		Inicio: 13/5/2015
Ident.	Título de la historia de usuario	Estimación
HU. 31	Un usuario puede cambiar su contraseña.	½
HU. 14	Un profesor o padre puede cambiar de estado una tutoría sin necesidad de entrar en la ficha de ésta.	1
HU. 32	Un usuario puede cambiar su imagen de perfil sin tener conexión a Internet.	3
HU. 30	Un usuario puede cerrar sesión.	½

Sprint 12		Inicio: 1/6/2015
Ident.	Título de la historia de usuario	Estimación
HU. 24	Un usuario puede ver un calendario con las tareas de una asignatura sin tener conexión a Internet.	1
HU. 33	Un usuario obtendrá las imágenes de los usuarios asociados.	3

5.2.2 Diagramas

En esta sección se mostrarán todos los diagramas correspondientes al desarrollo realizado, en primer lugar se mostrará un esquema que resume la idea de los tres niveles desarrollados en el proyecto: *back-end*, *middleware* y *front-end* y seguido de dicho esquema los diagramas entidad relación (E/R) de la base de datos tanto del servidor como los correspondientes al dispositivo Android.

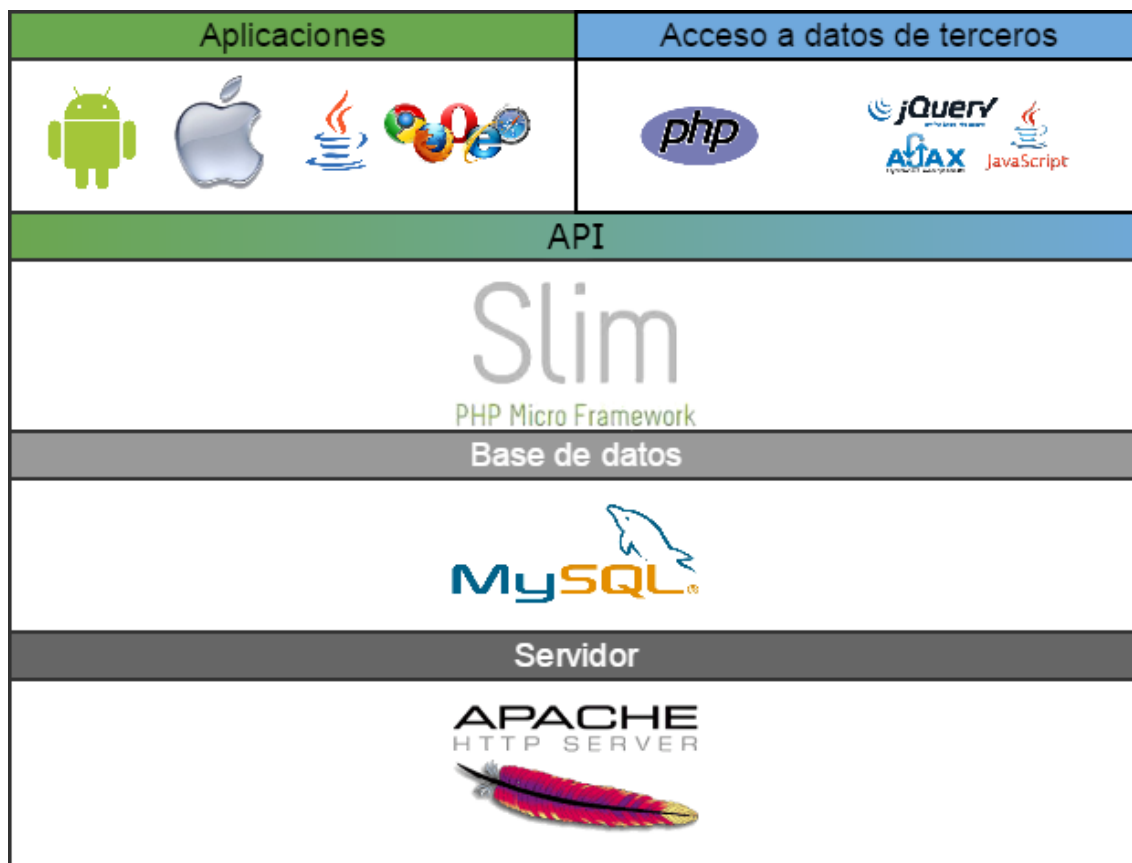


Ilustración 1 Esquema de niveles

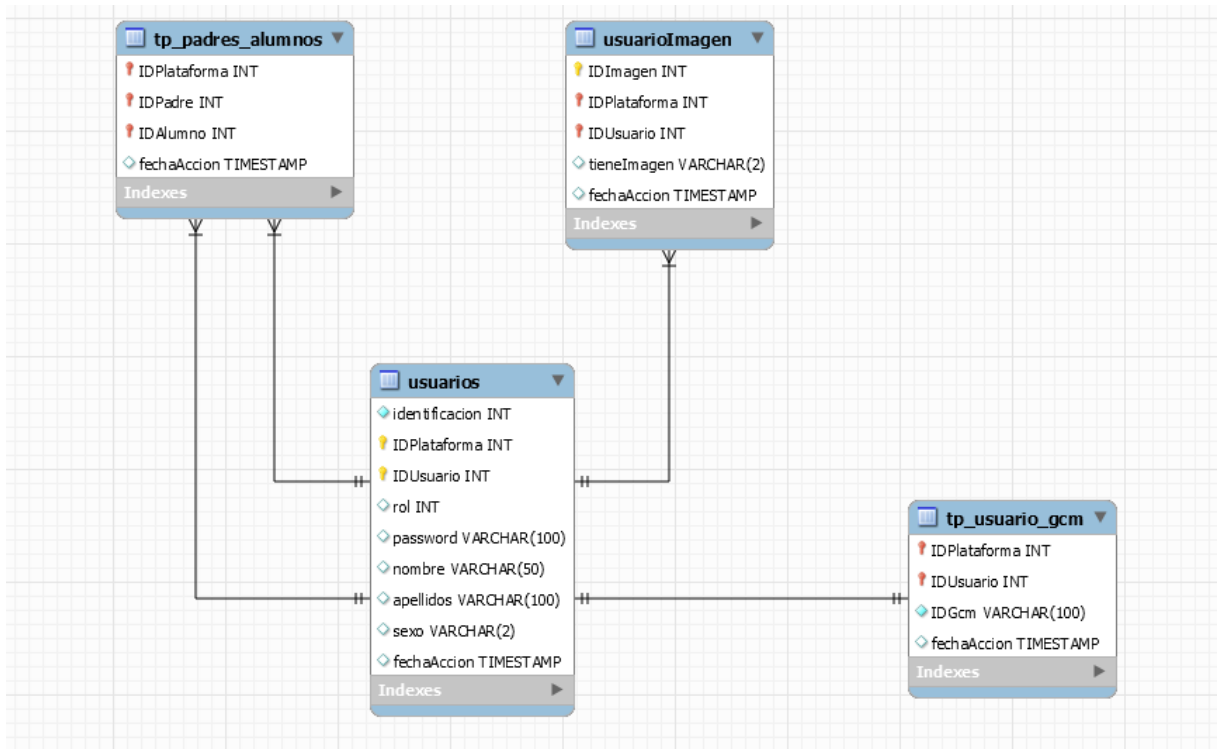


Ilustración 2 Relación de los usuarios (Servidor)

En la ilustración 25 se muestra la relación que guardan los usuarios entre sí, en primer lugar tenemos la relación padre alumno, que da lugar a la tabla *tp_padres_alumnos*, la cual no servirá para identificar los padres de un alumno o viceversa.

Para la gestión de imágenes se ha utilizado la tabla *usuarioImagen*, la cual mantiene actualizada el identificador de imagen de cada usuario para obtener su imagen.

Por último, para realizar las notificaciones *push* de *Google Cloud Messaging* (GCM), es necesario mantener la referencia del identificador que tiene un usuario en el servicio GCM con el identificador del servidor y la plataforma a la cual pertenezca el usuario.

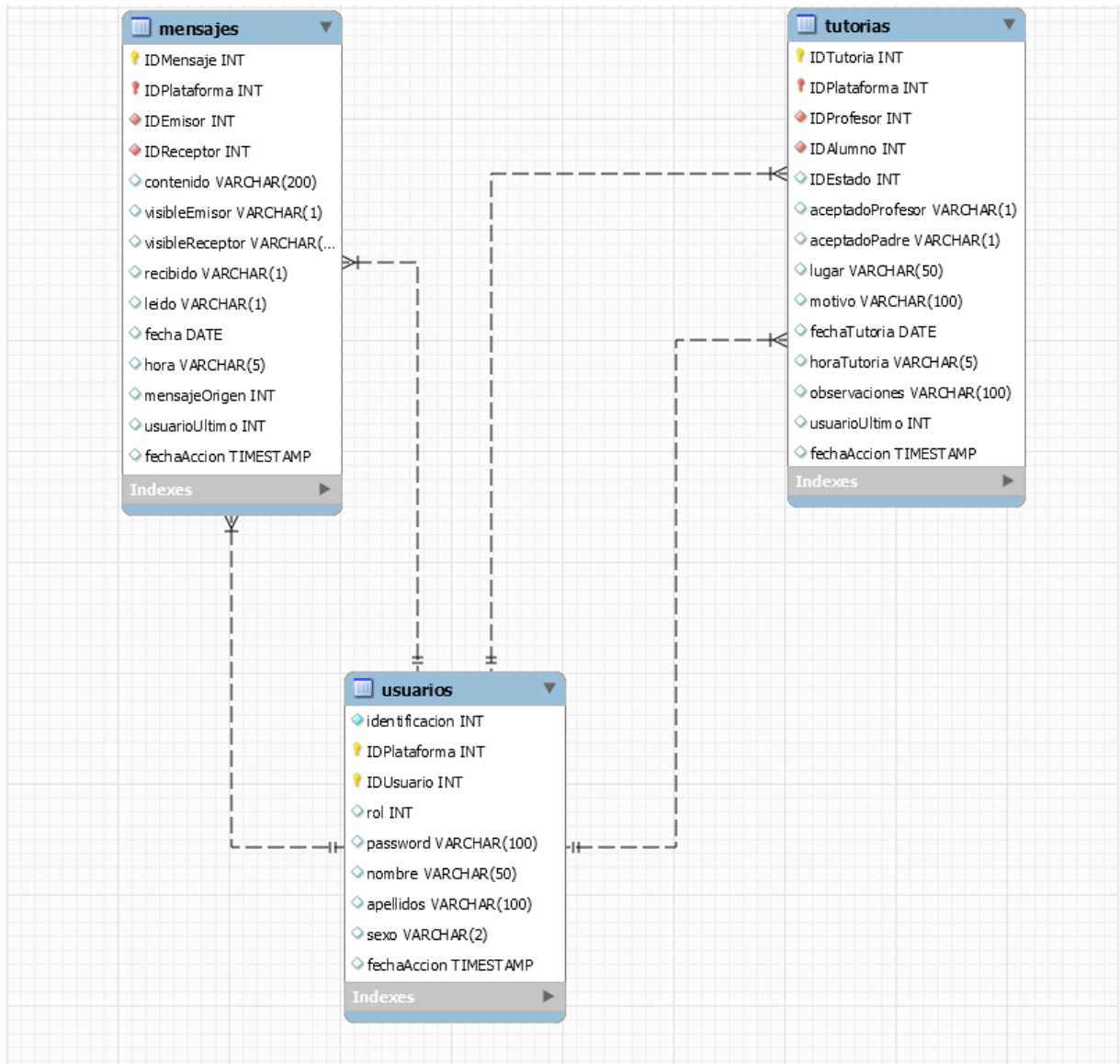


Ilustración 3 Relación de los mensajes y tutorías con los usuarios (Servidor)

Este diagrama representa los módulos de mensajes y tutorías, en los mensajes se puede ver la relación emisor-receptor que implica tener un identificador de ambos en la tabla *mensajes*, en tutorías ocurre de igual manera, tanto el profesor como el alumno crean la relación entre la tabla *usuarios* y *tutorías*.

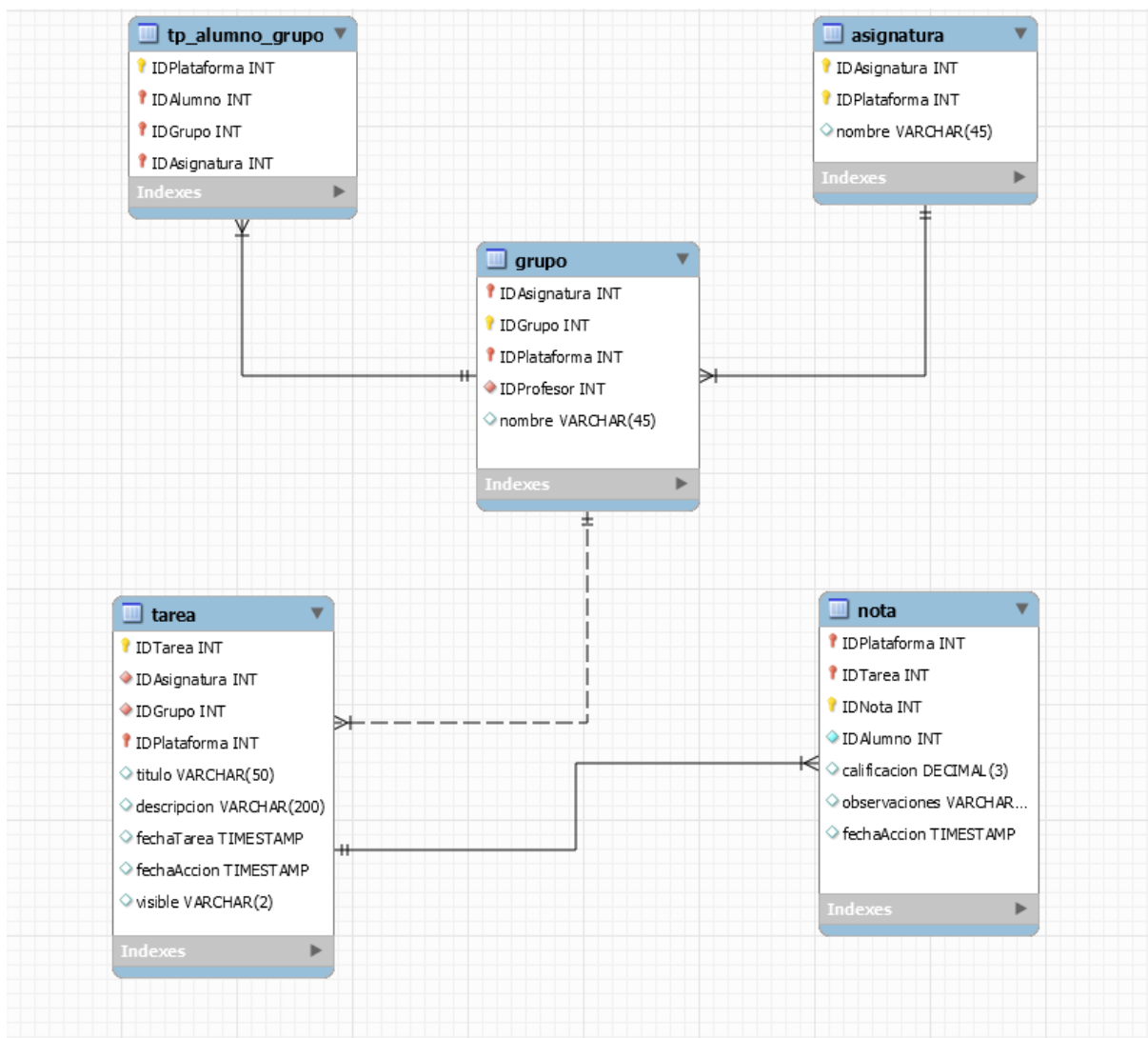


Ilustración 4 Relación entre el alumno, grupo, tarea y nota (Servidor)

Esta última relación podría decirse que es la académica, ya que mantiene la relación entre los alumnos, sus grupos y las tareas realizadas para dichos grupos.

Como se puede ver un grupo pertenece a una asignatura, pero no ocurre de igual manera al revés, por ello la tabla *grupo* debe tener como atributo la asignatura a la que pertenece, las tareas asignadas a dicho grupo llevarán el identificador del grupo en la tabla *tarea*, que en este caso sería el grupo y la asignatura de la que se tratase.

Para las calificaciones tan sólo es necesario el identificador de la tarea en la tabla *nota*.

Por último la tabla *tp_alumno_grupo* almacena la relación entre el usuario alumno y el grupo al que pertenece.

A continuación mostraré los diagramas E/R correspondientes a la base de datos Android, la cual es muy parecida a la base de datos del servidor, pero con algunas modificaciones en los campos de las tablas y la eliminación de algunas tablas.

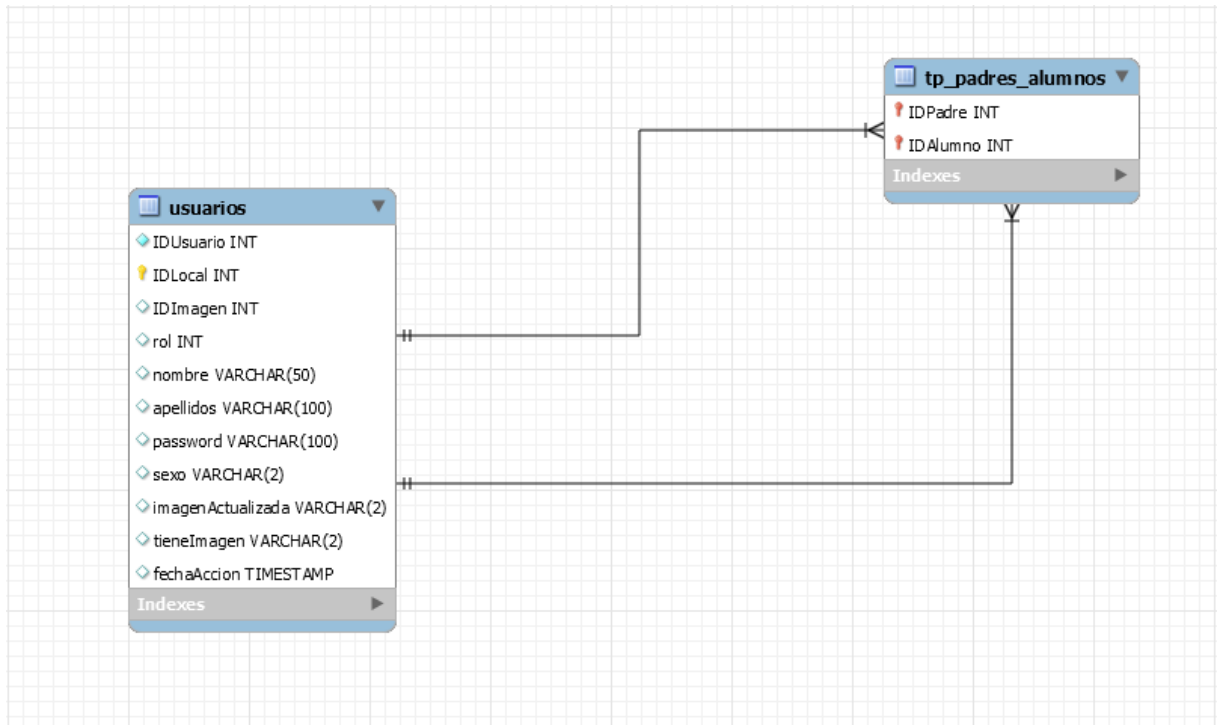


Ilustración 5 Relación de los usuarios (Android)

Al igual que en el servidor, tenemos una tabla para establecer la relación padre-alumno, la información correspondiente a la imagen del usuario está incluida en la tabla *usuarios* y la tabla para el servicio GCM desaparece, ya que esa información no es necesaria en el dispositivo Android.

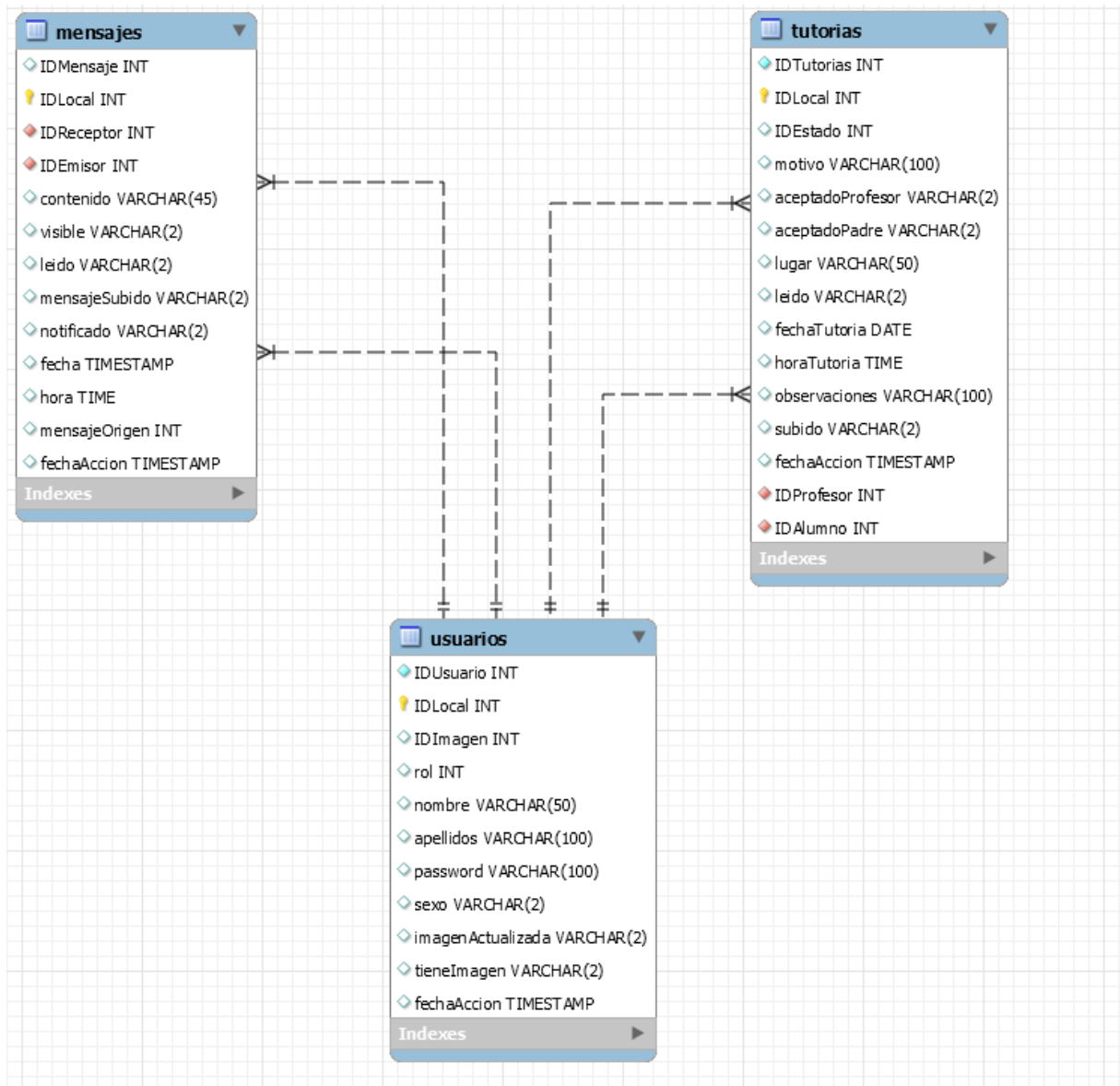


Ilustración 6 Relación de los mensajes y tutorías con los usuarios (Android)

En esta ocasión para la relación de mensajes y tutorías con los usuarios todas las tablas se mantienen, los únicos cambios son la inclusión de dos campos en la tabla *mensajes* y en la tabla *tutorías*, estos campos son *IDLocal* y *subido*, el primero nos indica el ID de cada objeto en la base de datos local, el segundo nos informa si la entrada está subida al servidor.

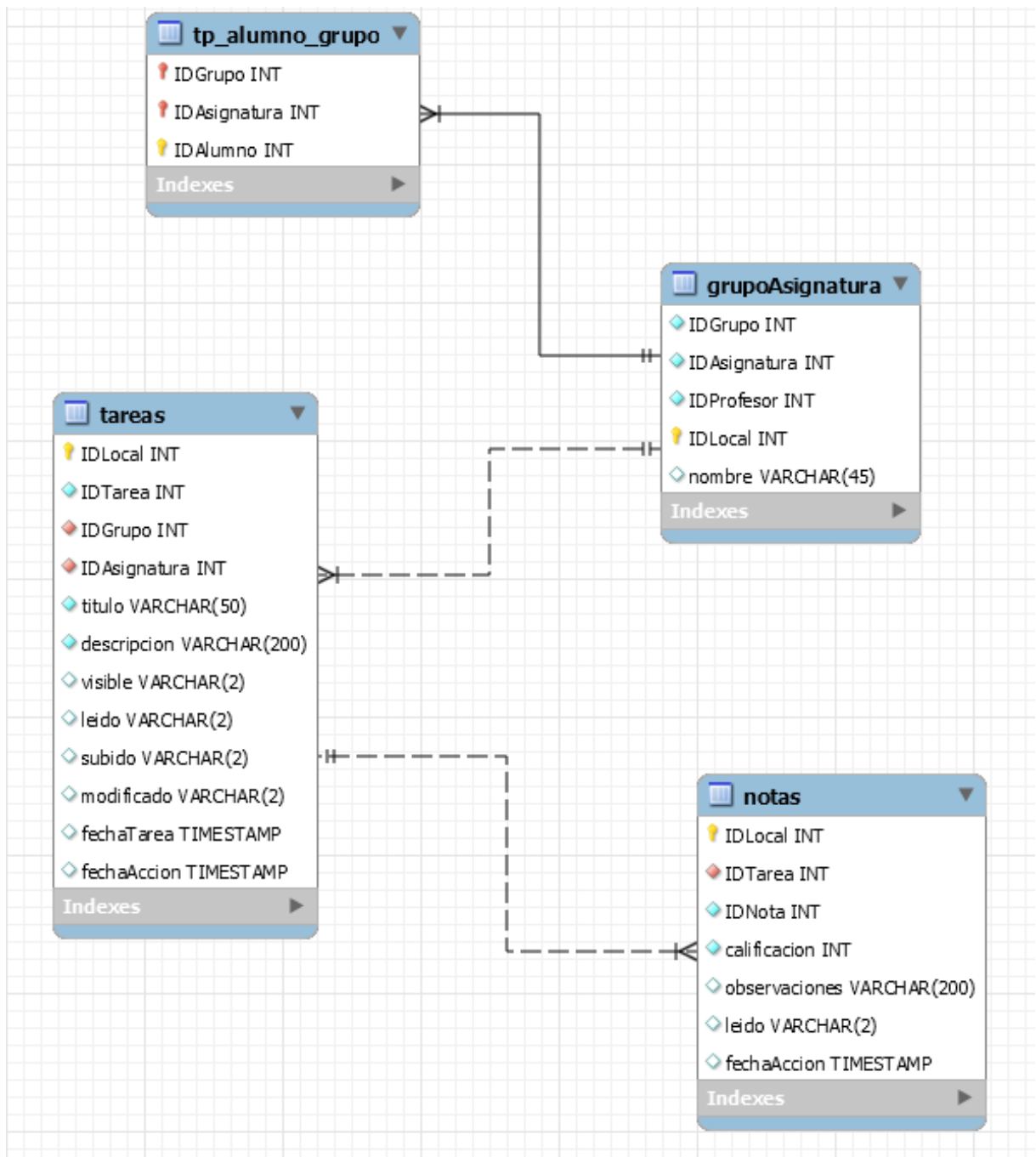


Ilustración 7 Relación entre el alumno, grupo, tarea y nota (Android)

Para reducir el tamaño de la base de datos, se ha fusionado la relación existente en la ilustración 27 entre las tablas *grupo* y *asignatura* y ha dado lugar a la tabla *grupoAsignatura*, el resto de tablas son iguales que en el servidor, añadiendo el identificador local para cada objeto de las tablas.

A continuación voy a mostrar el diagrama de clases de la aplicación Android en cuanto a los objetos utilizados representados en clases.

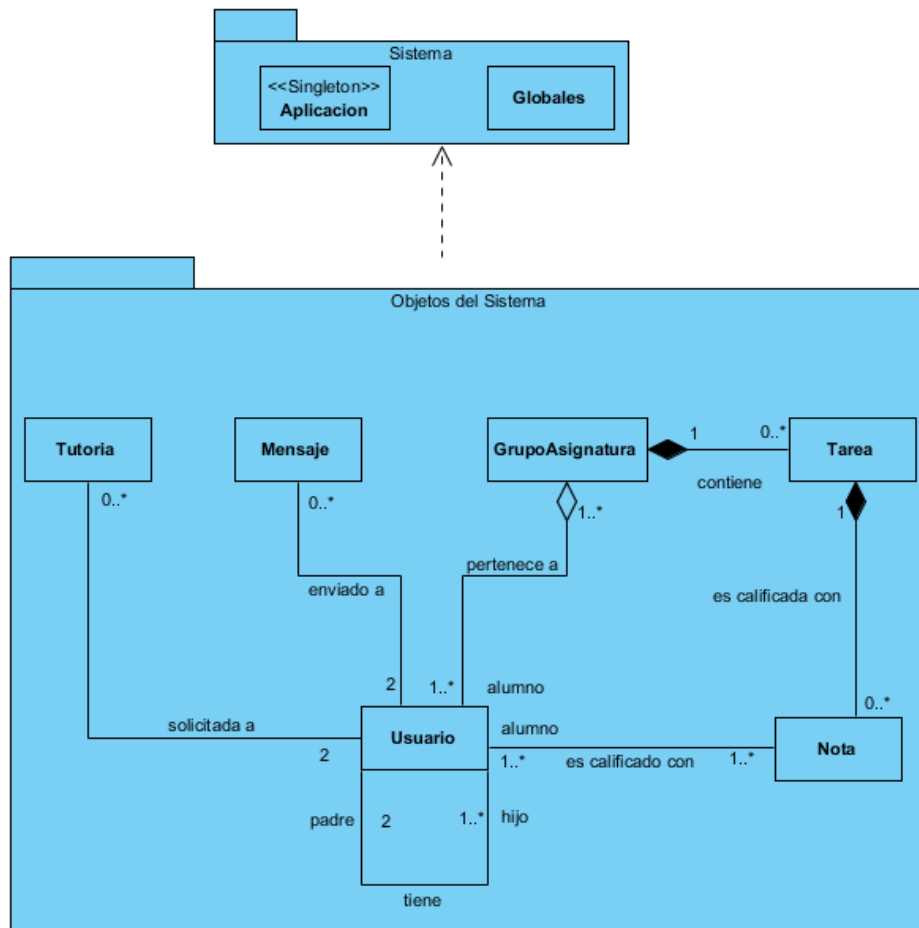


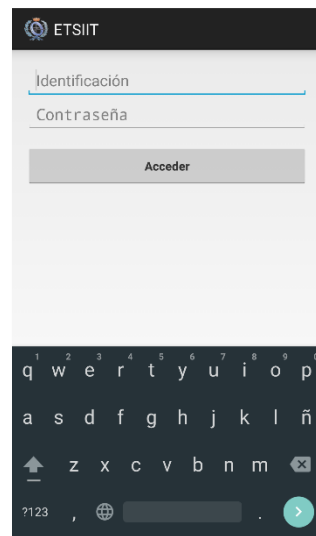
Ilustración 8 Diagrama de clases

En el diagrama podemos ver que el usuario es el principal actor de la aplicación, ya que tiene relación directa con casi todos los objetos del sistema, también se aprecia cierta similitud con los diagramas E/R de las bases de datos anteriormente mostrada.

En el paquete sistema tenemos las clases *Aplicación* y *Globales*, las cuales serán utilizadas por todos los objetos del sistema, ya que contiene variables de utilidad global.

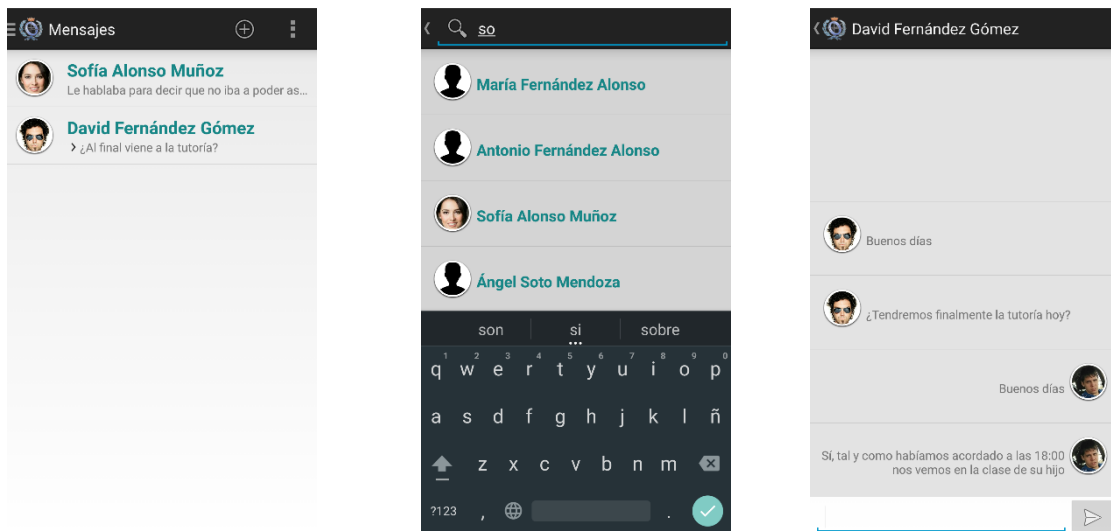
5.2.3 Interfaces de Usuario

La aplicación está compuesta por diferentes interfaces usuario, dichas interfaces son mostradas a continuación, y en caso de que proceda alguna aclaración o concepto necesario, aunque más tarde tanto los elementos de interacción como aquellos iconos que representen un concepto que se encuentren en la aplicación serán descritos y explicados.



9 Inicio de sesión

Pantalla de inicio de la aplicación, el usuario debe autenticarse y pulsar sobre acceder para comenzar la descarga de datos necesarios para el funcionamiento de la aplicación.

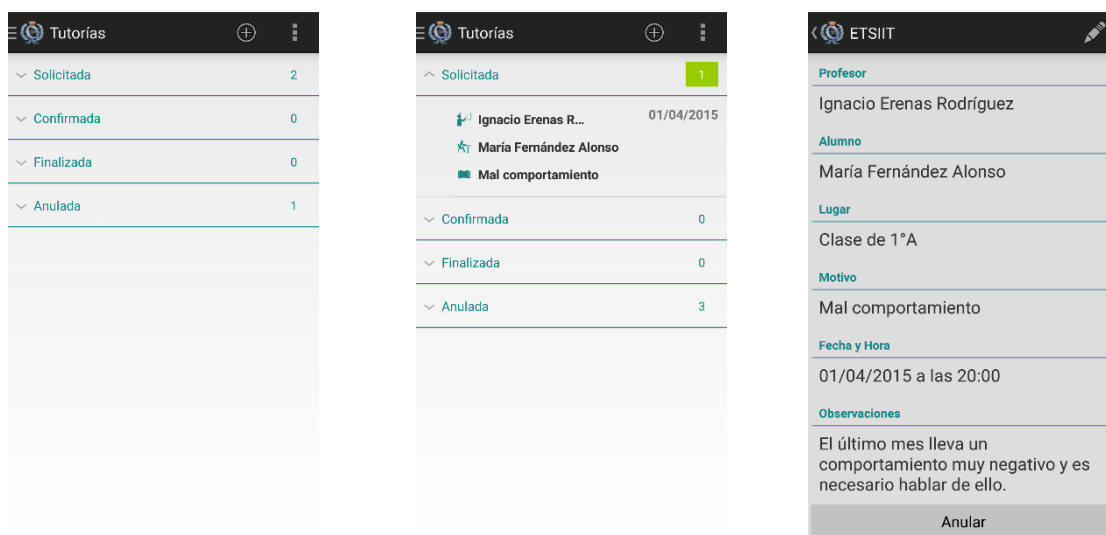


12 Buzón de entrada

11 Iniciar conversación

10 Conversación

Pantallas referentes a los mensajes, en la ilustración 12 podemos ver las últimas conversaciones del usuario, en la pantalla 11 la lista de usuarios filtrados para iniciar una nueva conversación y finalmente en la pantalla 10 la pantalla de conversación con el usuario.



13 Ficha de tutoría

15 Tutorías agrupadas

14 Tutorías novedades

ETSIIIT

Profesor
Ignacio Erenas Rodríguez

Alumno

Lugar

Motivo

Fecha Hora

Observaciones

18 Formulario nueva tutoría

ETSIIIT

Profesor
Ignacio Erenas Rodríguez

Alumno
ju

Juan León Carrasco

Juana Torres Arias

Fecha Hora

17 Formulario nueva tutoría II

ETSIIIT

Profesor
Ignacio Erenas Rodríguez

Alumno

30 may. 2014

01 jun. 2015

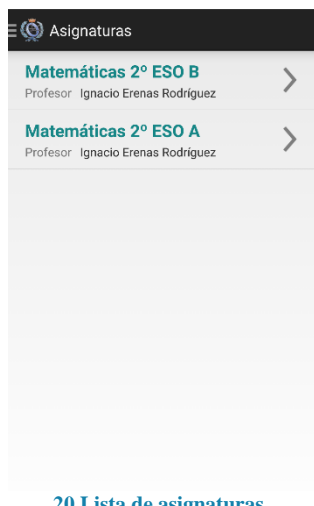
02 jul. 2016

Cancelar Aceptar

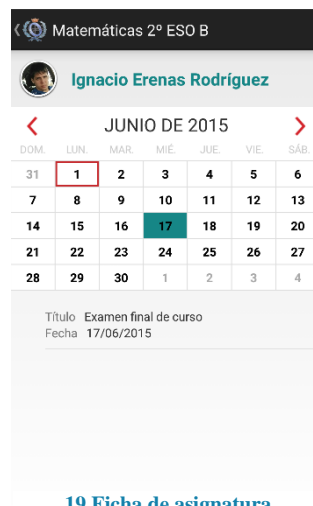
16 Formulario nueva tutoría III

Este grupo de pantallas representan el módulo de tutorías, en la pantalla 15 podemos ver las tutorías agrupadas en los 4 estados posibles, en la pantalla 14 hay un ejemplo de una tutoría no leída con un estado de tutorías desplegado.

La pantalla 13 muestra la ficha de una tutoría concreta con la información correspondiente, por último, las pantallas 16,18 y 19 representan el formulario de creación de una nueva tutoría y algunas de sus funcionalidades como el autocompletar del alumno (17) y la selección de fecha (16).

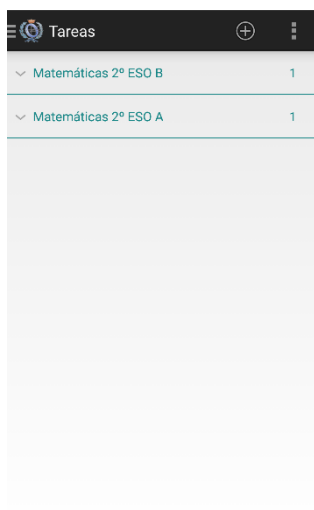


20 Lista de asignaturas

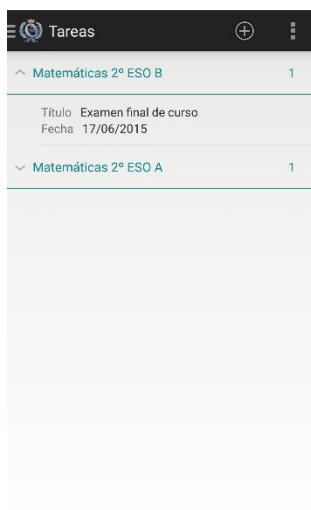


19 Ficha de asignatura

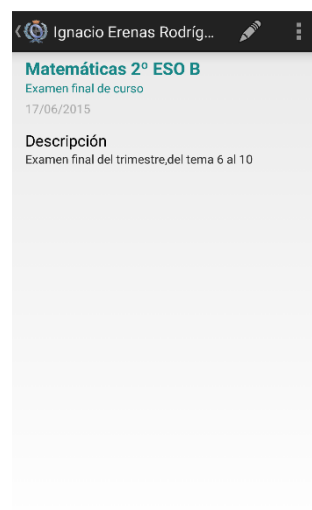
Estas imágenes nos muestran las pantallas referentes a las asignaturas, en la imagen 20 se muestra un listado de las asignaturas para el usuario del dispositivo o hijo seleccionado, en la 19 se muestra una ficha de la asignatura, en la cual se muestra el profesor impartidor, un calendario con las próximas tareas y finalmente las últimas tareas para dicha asignatura.



23 Lista de tareas agrupadas



22 Tareas con una asignatura desplegada



21 Ficha de tarea

ETSIT

Título de la Tarea

Examen Final

Asignatura

Matemáticas 2º ESO A

Fecha

09/06/2015

Descripción

Examen final 2º cuatrimestre

Calificaciones

Matemáticas 2º ESO A

Título Examen final

9.0

Alumnos

	Juan León Carrasco	--
	Adrián León Carrasco	8.0
	Excelentes las respuestas 2 y 3	
	Juana Torres Arias	--
	María Fernández Alonso	--
	Antonio Fernández Alonso	--

P

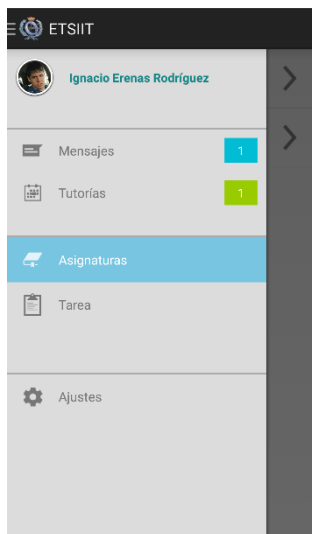
24 Formulario de tarea

25 Calificaciones de un alumno

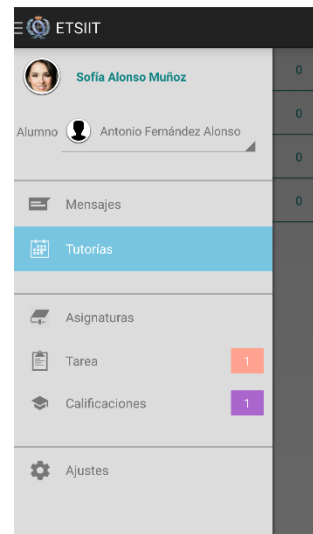
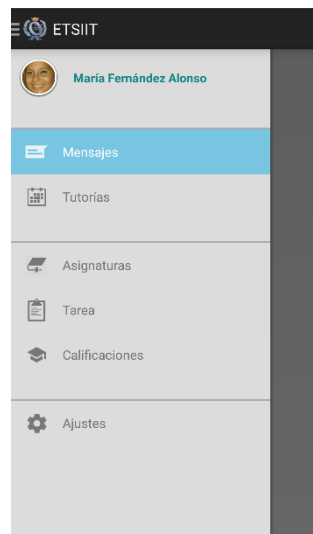
26 Alumnos relacionados con la
tarea

Pantallas de las tareas, al igual que en las tutorías, se tiene un listado agrupado (23) que pueden ser desplegadas (22) para seleccionar una tarea y ver su ficha (21).

La imagen 24 muestra el formulario a rellenar para crear una nueva tarea, y la imagen 26 muestra los alumnos que deben realizar la tarea junto con su calificación, mientras que la imagen 25 muestra las calificaciones para el alumno seleccionado.



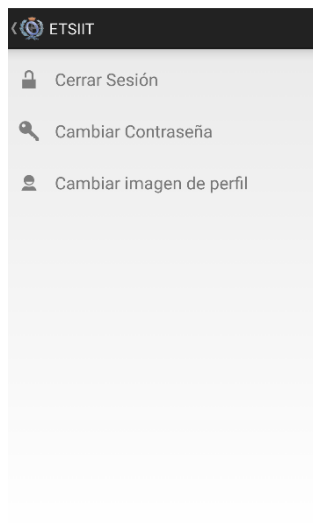
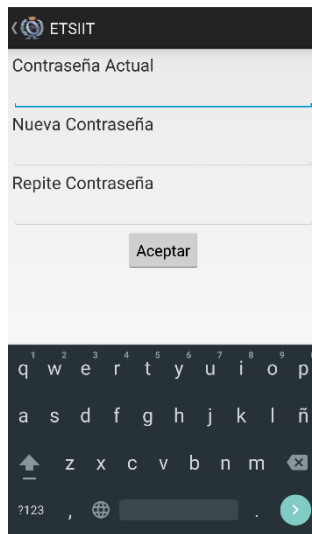
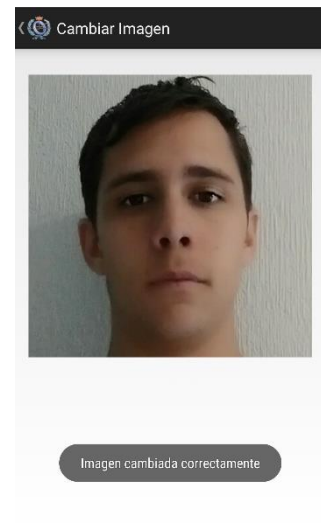
27 Menú navegador profesor



29 Menú navegador alumno


28 Menú navegador padre


En las tres pantallas anteriores vemos el menú de navegación de cada uno de los roles que pueden iniciar sesión en la aplicación, en la imagen 27 vemos el caso de un profesor con novedades en mensajes y tutorías, en la navegación del alumno (imagen 29) aparece la pestaña de calificaciones, y por último en el rol de padre (imagen 28) aparece el alumno que seleccionar sobre el cual ver las tareas y calificaciones de éste.


**32 Ajustes de la aplicación****31 Cambiar de contraseña****30 Cambiar imagen**


La imagen 32 nos muestra los ajustes de la aplicación, estos nos permite cambiar la contraseña actual (31) y la imagen de perfil que aparecerá para los demás usuarios (30).


En la aplicación, el usuario puede encontrar varios iconos que representan un concepto, a continuación muestro un listado, indicando el concepto que representan, una breve descripción, sus datos asociados y la representación icónica:


Concepto a Representar	Mensajes
Descripción	Icono para indicar que la información mostrada es acerca de mensajes.
Datos Asociados	Mensajes.
Representación	


Concepto a Representar	Tutorías
Descripción	Icono para indicar que la información mostrada es acerca de tutorías.
Datos Asociados	Tutorías.
Representación	

Concepto a Representar	Asignaturas
Descripción	Icono para indicar que la información mostrada es acerca de asignaturas.
Datos Asociados	Asignaturas.
Representación	


Concepto a Representar	Tareas
Descripción	Icono para indicar que la información mostrada es acerca de tareas.
Datos Asociados	Tareas.
Representación	


Concepto a Representar	Calificaciones
Descripción	Icono para indicar que la información mostrada es acerca de calificaciones.
Datos Asociados	Calificaciones.
Representación	


Concepto a Representar	Novedades
Descripción	Número con un color de fondo asociado al dato del cual hay el número de novedades que indica el número.
Datos Asociados	Mensajes, tutorías, tareas, calificaciones.
Representación	

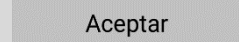
Concepto a Representar	Mensaje enviado
Descripción	Aspa mostrada en la pantalla de mensajes para indicar que el último mensaje ha sido enviado por el usuario.
Datos Asociados	Mensajes.
Representación	

A continuación haré un breve listado de cada elemento de interacción que el usuario pueda encontrar en la aplicación, este listado nos mostrará la representación de dicha interacción, un concepto asociado, una breve descripción y datos a los cuales está asociado dicha interacción.

Acción	Añadir
Objetivo	Icono para añadir un nuevo evento dependiendo del módulo en el cual se muestre.
Representación	

Acción	Mostrar menús
Objetivo	Icono para mostrar los menús asociados a la pantalla actual.
Representación	

Acción	Enviar mensaje
Objetivo	Icono para enviar el mensaje.
Representación	

Acción	Realiza la acción mostrada en el texto.
Objetivo	Botón para realizar la acción que indica su texto
Representación	

5.3 Implementación

Las herramientas utilizadas durante la realización del proyecto han sido varias y con diferentes propósitos, desde librerías para incluir nueva funcionalidad en la aplicación, hasta para mejorar la planificación del proyecto.

La herramienta online *Scrumf*⁴ se utiliza para la metodología de desarrollo ágil, y cuenta con todas las herramientas necesarias para monitorizar el avance del proyecto.

Para el código tanto en Android como en PHP se utilizó el control de versiones Subversion, alojado en la página xp-dev.com, de esta forma tras cada sprint realizado se subía una nueva versión de la aplicación.

A continuación describiré herramientas utilizadas tanto en la aplicación *Android* como en el código PHP del servidor.

5.3.1 Android

La programación en Android se realizó utilizando el lenguaje de programación Java, utilizando el ambiente de desarrollo integrado (en inglés *Integrated development environment*; IDE) Android Studio⁵, el cual también contenía herramientas para la creación de las distintas interfaces que componen la aplicación, además de la creación de la documentación del código Java con la utilidad Javadoc.

Los lenguajes de programación nativos del sistema operativo Android, son C y Java, el hecho de haber escogido Java viene dado por los siguientes motivos:

- Mayor conocimiento del lenguaje y dominio gracias a la práctica adquirida durante el transcurso del Grado.
- Mayor rendimiento del sistema frente a C.
- Más librerías para añadir funcionalidad extra a la aplicación.
- Variedad de IDEs para la creación de aplicaciones con el lenguaje Java.

Para la gestión de la base de datos en el dispositivo Android se ha utilizado SQLite, a pesar de estar escrita en C se puede utilizar en Java gracias al driver “*SQLite JDBC*”.

Algunas funcionalidades de la aplicación se han conseguido gracias a librerías externas, dichas librerías son repositorios maven los cuales son añadidos gracias a la herramienta de construcción Gradle.

Dichas funcionalidades son:

1. Calendario que encontramos en la ficha de una asignatura, dicha funcionalidad ha sido gracias a la librería *Caldroid*⁶.
2. La forma redondeada de las imágenes de los perfiles de usuario se obtiene de la librería *CircularImageView*⁷.
3. La funcionalidad de arrastrar y refrescar en el buzón de entrada en los mensajes se consigue con la librería *Action-Bar Pull To Refresh*⁸.

5.3.2 Servidor

El servidor instalado contiene el sistema de infraestructura WAMP, dicho acrónimo proviene de las herramientas que utiliza:

- Windows, como sistema operativo.
- Apache, como servidor web.
- MySQL, como gestor de base de datos.
- PHP, como lenguaje de programación, aunque también se puede utilizar Perl o Python.

El haber escogido WAMP tiene dos razones fundamentales, en primer lugar la familiaridad con cada una de sus herramientas ya que tanto Apache como MySQL y PHP han sido herramientas de trabajo habituales durante los cuatros años de transcurso del grado.

En segundo lugar la idoneidad de las herramientas para el proyecto a desarrollar, ya que MySQL es una de las bases de datos de código abierto más utilizadas, Apache tiene una fácil configuración pero una potente seguridad.

Para potenciar la programación en PHP se ha utilizado el IDE *PhpStorm*¹⁹, ya que esta herramienta ofrece funcionalidades como depuración del código o conexión con la base de datos con la cual se trabaja.

El software *MySQL Workbench*¹⁰ ha permitido el diseño, creación y modificación de la base de datos MySQL, además ofrece potentes herramientas como la ingeniería inversa o reversa para la creación de la base de datos a partir de diagramas o viceversa.

La API que facilita la conexión de la base de datos con el resto de aplicaciones se ha realizado con el framework *Slim*¹¹.

6. CONCLUSIONES Y VÍAS FUTURAS

6.1 Académica

Podemos concluir que se han adquirido las competencias que otorga la realización del Grado en Ingeniería Informática, esto queda demostrado en la utilización de distintas tecnologías para el software desarrollado (PHP, Java, SQL...), cuyo conocimiento ha sido adquirido con la realización de las distintas asignaturas del Grado.

También queda reflejado en el análisis, planificación y desarrollo llevado a cabo el conocimiento obtenido fruto de la especialización en Ingeniería del Software, de forma que para llevar a cabo este proyecto se ha hecho uso de los distintos conocimientos obtenidos tanto en la parte general del desarrollo del Grado, como de la parte específica.

6.2 Profesional

El *software* desarrollado tiene dos grandes ventajas desde una perspectiva profesional, en primer lugar ha sido desarrollado mediante una metodología de desarrollo ágil, la cual disminuye el esfuerzo a realizar en caso de necesitar algún cambio.

En segundo lugar el contar con una API consigue que el desarrollo de cualquier aplicación en cualquier plataforma sea posible de una manera sencilla y eficaz.

De forma que la reducción de esfuerzo que conlleva estas dos ventajas, hace que se pueda implantar esta plataforma de apoyo a la docencia en cualquier instituto y sus costes no sean muy elevados.

7. BIBLIOGRAFÍA

1 Moodle - <https://moodle.org/> - 2015

2 Seneca - <http://www.juntadeandalucia.es/educacion/portalseneca/web/seneca/inicio> - 2014

3 [La sociedad española, la más “conectada” de Europa: 8 de cada 10 móviles son smartphones y el 78% de los internautas son usuarios intensivos.](#)

4 Scrumrf - <http://scrumrf.com/> - 2014

5 Android Studio - <https://developer.android.com/sdk/index.html> - 2014-2015

6 Librería Caldroid - <https://github.com/roomorama/Caldroid> - Licencia MIT - 2014

7 Librería imágenes de perfil redondas - <https://github.com/googolmo/CircularImageView> - Licencia Creative Commons 4.0 - 2014

8 Librería Pull to Refresh - <https://github.com/chrisbanes/ActionBar-PullToRefresh> - Apache 2.0 - 2014

9 PhpStorm - <https://www.jetbrains.com/phpstorm/> - 2014

10 MySQL Workbench - <https://www.mysql.com/products/workbench/> - 2014

o, **Ignacio Erenas Rodríguez**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 76654227-B, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.



Fdo: **Ignacio Erenas Rodríguez**

Granada a 1 de julio de 2015.

D. Luis Castillo Vidal, Profesor del Departamento de Ciencias de la Computación e I.A. de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado **E-duca+**: Plataforma de Apoyo a la Docencia, ha sido realizado bajo su supervisión por **Ignacio Erenas Rodríguez**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 1 de julio de 2015.

El director:

Luis Castillo Vidal

Agradecimientos

A mi familia y amigos.

*A todas aquellas personas que de forma directa o indirecta han hecho
posible que llegue hasta aquí.*

D. Luis Castillo Vidal, Profesor del Departamento de Ciencias de la Computación e I.A. de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado

E-DUCA+: PLATAFORMA DE APOYO A LA DOCENCIA

ha sido realizado bajo su supervisión por

Ignacio Erenas Rodríguez

y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe

en Granada a 1 de julio de 2015.

El director:



Luis Castillo Vidal