



**Universidad de Valladolid**

**E. U. DE INFORMÁTICA (SEGOVIA)**

**Grado en Ingeniería Informática de Servicios y  
Aplicaciones**

---

**AE- DROID**

---

***Alumno: Moisés Martínez Mateu***

**Tutor: Fernando Díaz Gomez**



**Tabla de contenido**

SECCIÓN 1: MEMORIA DEL PROYECTO..... 5

    Estructura de los Entregables ..... 5

    CAPÍTULO 1: Aplicación AE-DROID ..... 7

        Introducción..... 8

        Descripción General ..... 8

        Documentación Previa..... 10

        Motivación ..... 14

    CAPÍTULO 2: PLANIFICACIÓN DEL PROYECTO ..... 16

        Introducción..... 17

        Descripción ..... 17

        Tecnología Empleada..... 17

    Objetivos ..... 18

        Identificación de los Stakeholder ..... 19

        Modelo de proceso para el desarrollo ..... 20

    Entorno Operativo y Arquitectura..... 22

        Arquitectura Física..... 22

        Arquitectura Lógica..... 23

        Herramientas Empleadas..... 31

        Hardware ..... 31

    Estudio de Viabilidad ..... 33

        Viabilidad Tecnológica ..... 33

        Metodología Empleada..... 41

    CAPÍTULO 3: ANALISIS DEL SISTEMA..... 44

        Catálogo de requisitos del sistema ..... 47

        Requisitos Funcionales ..... 50

        Diagramas de Casos de Uso ..... 52

        Requisitos No funcionales..... 64

        Diagrama de Clases ..... 67

        Diagrama de Secuencia ..... 75

    CAPÍTULO 4: DISEÑO DEL SISTEMA..... 76

        Diagrama de Clase..... 78

        Clase Actividad..... 80

        Clase Alumno..... 83

        Clase Medición ..... 87

        Clase Observacion ..... 90

Diseño General del Sistema.....	93
Diseño de la Base de Datos.....	94
Diseño de las interfaces de gráficas.....	104
CAPÍTULO 5. PRUEBAS DEL SISTEMA .....	110
Pruebas y Validación.....	111
Pruebas Unitarias.....	112
Pruebas de Integración.....	112
Prueba del Sistema.....	112
Pruebas de Aceptación.....	113
Alcance de las Pruebas .....	115
Criterios de Aceptación o Rechazo.....	119
CAPÍTULO 6. CONCLUSIONES PERSONALES.....	120
CAPÍTULO 7. FUTURAS MEJORAS .....	121
CAPÍTULO 8. DEFINICIONES.....	122
SECCION 2. MANUAL DE USUARIO.....	126
Manual de Instalación.....	127
Manual de Desinstalación.....	130
Manual de Usuario.....	132
BIBLIOGRAFIA .....	143
OTROS RECURSOS.....	143

## **SECCIÓN 1: MEMORIA DEL PROYECTO**

### **Estructura de los Entregables**

#### Identificación del TFG

Título: AE-DROID

Autor: Moisés Martínez Mateu

Tutor: Fernando Díaz Gomez

#### Organización de la Memoria

La documentación de este Trabajo Fin de Grado contendrá la siguiente estructura:

1. Sección PRIMERA: MEMORIA DEL PROYECTO, en este apartado contendrá toda la información de la memoria. Se divide en los siguientes capítulos:

- a. Capítulo 1: Aplicación AE-DROID
- b. Capítulo 2: Planificación
- c. Capítulo 3: Análisis del Sistema
- d. Capítulo 4: Diseño del Sistema
- e. Capítulo 5: Pruebas del Sistema
- f. Capítulo 6: Conclusiones Finales
- g. Capítulo 7: Futuras Mejoras
- h. Capítulo 8: Definiciones

2. Sección SEGUNDA: MANUAL DE USUARIO, en este apartado se abordará toda la información relacionada con el manual técnico del manejo, así de como su instalación.

- a. Capítulo 1: Manual de Instalación
- b. Capítulo 2: Manual de Desinstalación
- c. Capítulo 3: Manual de Usuario

3. Sección TERCERA: OTROS RECURSOS , en este apartado se expondrán aquellos recursos utilizados para el desarrollo de este proyecto:

- a. Capítulo 1: Bibliografía
- b. Capítulo 2: Software

### Estructura del CD

Dentro de los entregables, se encuentra el CD con la documentación en formato electrónico, así como los ficheros ejecutables y el código fuente al completo.

- CD\_TFG\_AE-DROID
  1. Documentación: en este directorio se encontrara la documentación del proyecto en formato pdf.
  2. Software: En este directorio es donde se encuentra el SW de la aplicación, y se divide:
    - Ejecutable: donde se encuentra el fichero APK
    - Código fuente: aquí se encuentra el código fuente de la aplicación
    - El auto instalador XAMPP para montar rápidamente un servidor web con base de datos
    - BD: se encuentra el fichero .sql para importar en la base de datos
    - Servidor: se encuentra el directorio htdocs que tiene que ir a la raíz del servidor.

# **CAPÍTULO 1: Aplicación AE-DROID**

## Introducción

La realización de este proyecto tiene como objetivo analizar el entorno educativo de hoy en día, para desarrollar gracias a las nuevas tecnologías, nuevas formas de mejorar el seguimiento de alumnos con necesidades especiales.

Porque tan importante es saber enseñar cada día nuevos contenidos como diagnosticar con mayor eficacia el rendimiento de cada alumno. Por ello, los objetivos de este proyecto es conseguir desarrollar una plataforma que simplifique las tareas comunes sobre el seguimiento de la enseñanza y los automatice.

Es importante mencionar que nos enfrentamos a una tarea más allá del simple diseño y desarrollo de un sistema, hay que comprender bien las áreas que se va a trabajar como por ejemplo como medir las necesidades especiales de los alumnos.

Una vez se ha estudiado en qué consisten los principales problemas que hemos de afrontar en la consecución de este proyecto. Se prosigue con el estudio y diseño del sistema final, en este caso, para dispositivos móviles.

## Descripción General

El sistema tiene como objetivo la captura de datos recogidos por los especialistas para el seguimiento del rendimiento de los alumnos.

Estos datos pueden ser de diversos tipos, durante el proyecto utilizaremos aquellos que permitan automatizar las hojas de seguimiento reales que se realizar a diario.

Los datos recogidos servirán para aplicar una serie de estímulos a aquellos alumnos con problemas para realizar ciertas actividades.

Posteriormente se realiza el seguimiento viendo en cada registro, que alumnos de un determinado grupo tiene problemas con ciertas actividades a través de las medidas recogidas.

Este proyecto permitirá acelerar el proceso de seguimiento de los alumnos y mejorar la calidad actual haciendo los procesos de análisis y tratamiento mucho más veloces.

Como ejemplos vemos alguna de las capturas de informes y otros formularios que se utilizan para el seguimiento de los alumnos



CONTROL ESFÍNTERES			AAA2	MES: OCTUBRE		
SEMANA	VAMOS-AL-BAÑO	PAÑAL-SECO	%SECO	HACE-PIS-WC	% HACE-PIS-WC	
1	15	3	20,00%	10	66,67%	
2	16	2	12,50%	8	50,00%	
3	12	5	41,67%	7	58,33%	
4	18	7	38,89%	7	38,89%	
5	14	4	28,57%	8	57,14%	

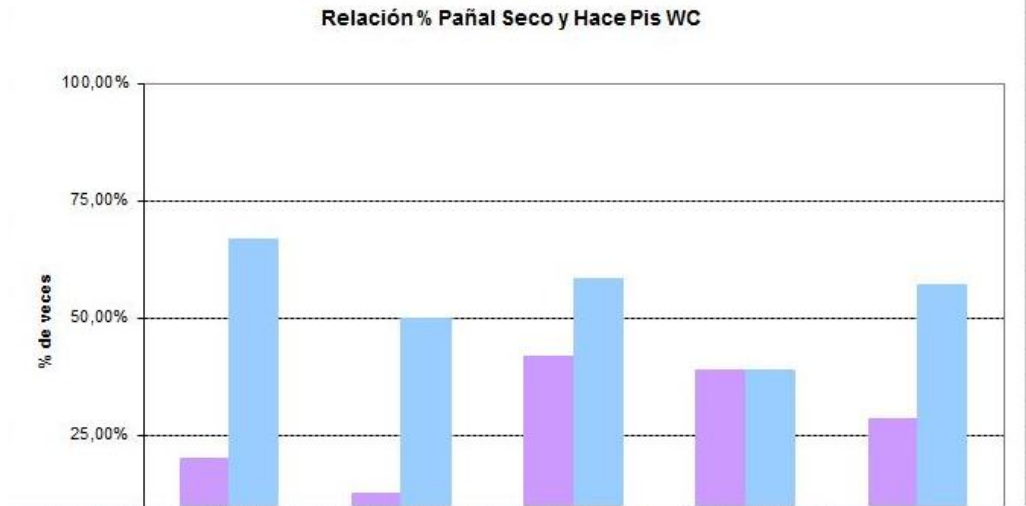


Ilustración 1- Informe sobre control de esfínteres en niños

PROGRAMA DE CONTROL DE ESFÍNTERES

Alumno: \_\_\_\_\_ Grupo: INFANTIL (1º B) Mes: OCTUBRE

\*M: Pañal Mojado / P: Hace Pis / C: Hace Caca / N: No hace Nada

	LUNES	MARTES	MÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO	
	Horas M P C N	Horas M P C N	Horas M P C N	Horas M P C N	Horas M P C N	Horas M P C N	Horas M P C N	
1ª Semana	12:15 SI SI 13:15 NO SI 14:15 NO SI 15:30 SI SI	9:45 SI SI SI 11:15 NO SI 12:45 NO SI 15:30 SI SI	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 NO SI	9:45 SI SI 11:15 NO SI 12:45 SI SI 15:30 SI SI	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 SI SI	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 NO SI		
2ª Semana	NO VIENE (este enfermo)	9:45 NO SI 11:15 SI SI 12:45 NO SI 15:30 SI SI	NO VIENE (tiene médico)	9:45 NO SI 11:15 NO SI 12:45 SI SI 15:30 NO SI	9:45 NO SI 11:15 NO SI 12:45 SI SI 15:30 NO SI	9:45 NO SI 11:15 NO SI 12:45 SI SI 15:30 NO SI		
3ª Semana	FIESTA	FIESTA	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 NO SI	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 NO SI	9:45 SI SI 11:15 NO SI 12:45 NO SI 15:30 NO SI			
4ª Semana	9:45 NO SI 11:15 NO SI 12:45 SI SI 14:20 NO SI 15:30 NO SI	9:45 NO SI 11:15 SI SI 12:45 NO SI 14:15 NO SI 15:30 NO SI	9:45 NO SI 11:15 NO SI 13:15 SI SI 14:30 NO SI 15:20 NO SI	9:45 NO SI 11:15 NO SI 12:45 NO SI 14:15 SI SI 15:20 NO SI	9:45 NO SI 11:15 SI SI 12:45 NO SI 14:15 SI SI 15:30 NO SI			
Observaciones	NO VIENE A LA HORA 12:45 SI SI 15:30 SI SI	11:45 NO SI 12:45 SI SI 14:30 SI SI 15:30 SI SI	10 SI SI 12 NO SI 13:45 SI SI 15:30 SI SI	9:45 SI SI 11:45 NO SI 13:45 SI SI 15:30 NO SI				

Ilustración 2-Seguimiento de Control de Esfínteres

## Documentación Previa

Para comprender la base fundamental de este proyecto, hay que buscar las leyes orgánicas que rigen sobre la educación. En este caso, según la ley orgánica 2/2006 establece los principios y fines de la educación, los cuales quedan resumidos en los siguientes apartados:

- a) La calidad de la educación es para todo el alumnado, independientemente de sus condiciones y circunstancias
- b) La equidad, que garantice la igualdad de oportunidades, la inclusión educativa y la no discriminación y actúe como elemento compensador de las desigualdades personales, culturales, económicas y sociales.
- c) La transmisión y puesta en práctica de valores que favorezcan la libertad personal, la igualdad, así como que ayuden a superar cualquier tipo de discriminación
- d) La flexibilidad para adecuar la educación a la diversidad de aptitudes, intereses, expectativas y necesidades del alumnado

De esta forma se reconoce una sola educación, validas en sus objetivos para todos los niños y niñas, con independencia de algunas dificultades o limitaciones que estos pueden tratar.

Estos cambios estriban en la introducción del **concepto de Necesidades Educativas Especiales (NEE)**, término implantado en **el informe de Warnock (1978)**, en el que se recomendaba la escolarización del alumnado con deficiencias en aulas ordinarias, interviniendo en contextos normalizados y cooperando con profesionales que debían describir y evaluar las necesidades educativas el alumnado, así como establecer un plan de acción o intervención entre ellos.

Partiendo del principio que todos los niños precisan en algún momento de su vida escolar, una serie de ayudas pedagógicas que afiancen la consecución de los fines de su educación, las necesidades especiales se refieren a aquellos alumnos que necesitaran de forma complementaria otras ayudas menos usuales.

### Evaluación sobre la educación especial

La Evaluación de las Necesidades Educativa Especiales (NEE) en el marco de las Adaptaciones Curriculares, es un tema amplio y complejo. Por ello, nos fijamos como objetivo prioritario clarificar ciertos contenidos que aún están difusos dentro de la Educación Especial.

Las Adaptaciones Curriculares en la actualidad parten de la propuesta realizada para todos los alumnos y requiere la coordinación de todos los profesionales, mientras que tradicionalmente el programa individual plantea unos objetivos, contenidos y actividades al margen del aula y participa únicamente el especialista.

Partiendo de este concepto, actualmente se necesita un modelo que plante unos objetivos educativos especializados a cada alumno, de forma objetiva e independiente del grupo.

Esta nueva evaluación sigue una serie de pautas:

- a) Orientada a la autoevaluación.
- b) Que sirva de vehículo informativo a todos los que participan en el proceso educativo.
- c) Que abarque todos los aspectos del desarrollo.
- d) Que se considere una actividad retroalimentadora de los procesos de E/A y facilitadora de cambios.
- e) Que tenga un carácter continuo.
- f) Que este dirigida a los procesos generados o por generar.

Estas pautas no pretenden solo comprobar los resultados, sino valorar de forma más objetiva cada progreso de cada alumno, donde se encuentran las limitaciones y evitarlas, fomentando el éxito del alumno durante el proceso. Cada alumno tiene sus propias características y hay que realizar una evaluación independiente sobre cada una de ellas.

Esto conllevaría a conjugar dos ramas sociales: la **educación** y la **psicología**, pero en nuestro proyecto este debate está lejos de nuestro alcance.

El problema está en cómo se puede conjugar estos dos aspectos, en algo tan concreto como la evaluación personalizada de cada alumno. Existen métodos y procedimientos que aúnan ambos enfoques.

### ¿Cómo evaluar?

El proceso de valoración de n.e.e. debe plantearse como un proceso multidisciplinar y multitarea que tiene como finalidad aportar la mayor cantidad posible de datos relevantes para la toma de decisiones. El método a utilizar depende de lo que se quiera evaluar. Una verdadera evaluación educativa debe combinar ambos métodos. Así señalados:

1. **Enfoque cualitativo:** Se caracteriza por defender una aproximación fenomenológica a los procesos educativos por estar interesado en comprender los fenómenos desde la perspectiva de quienes están implicados en ellos.

2. **Enfoque cuantitativo:** Busca las relaciones estables entre variables que permiten detectar qué circunstancias y fenómenos son causa o consecuencia de tales otros. Este defiende la utilización de métodos objetivos de recogida y análisis de datos: observación sistemática y controlada, uso de pruebas estandarizadas, etc. En general, métodos fiables.

Y algunas de las metodologías, cuya información utilizada para tomar decisiones sobre dónde, qué y cómo enseñar, así como las áreas fuertes y débiles del alumno, se obtiene de tres fuentes generales: Observaciones, tests e impresiones (Salvia y Hughes, 1990):

**-Observación:** Es un procedimiento evaluativo que implica escuchar o mirar a una persona así como hacer juicios sobre su rendimiento. Para que se convierta en instrumento de evaluación "debe ser intencional frente a lo cotidiano y casual", y su finalidad reside en recoger información con un fin concreto, y captar el significado de una conducta, libremente escogida, evitando su manipulación. Según el grado de estructuración, puede realizarse de manera espontánea sobre las conductas naturales de uno o varios alumnos, o en situaciones predeterminadas de manera que se incremente la conducta de un alumno.

Generalmente se propone el uso combinado de los dos tipos de observaciones, para que la evaluación pueda ser más completa.

**-Tests:** Son un modo de obtener observaciones sistemáticas en situaciones estructuradas o tipificadas con la finalidad de describir la conducta del alumno. Debemos diferenciar entre: **Tests publicados con o sin norma.**

- Los tests con norma son tests que comparan la ejecución de un estudiante con respecto a las ejecuciones de los estudiantes similares.
- Los tests sin normas para ser interpretados suelen tener utilidad para facilitar el trabajo de construcción de tests por parte de los profesores.
  1. Tests referidos a criterio: comparan la puntuación del alumno con un criterio específico. En estos procedimientos es importante la tarea, y se establece el criterio para considerarla superada por cualquier alumno.
  2. Tests referidos a un objetivo: aquellos referidos a una norma que tienen objetivos instruccionales escritos para los ítems de un tests individual.
  3. Tests de medida: contienen un elevado número de ítems para permitir adecuadas discriminaciones entre los sujetos.
  4. Tests diagnóstico: contienen un número suficiente de ítems como para permitir realizar inferencias correctas sobre el tipo de errores que un alumno comete o sobre el dominio de un estudiante respecto a unas habilidades determinadas.

**-Impresiones:** Son generalmente observaciones no sistemáticas, por tanto menos estructuradas que las anteriores. Las impresiones son útiles en los momentos iniciales de la evaluación. Entre ellas destacan: Escalas de valoración, instrumentos que requieren que realicen juicios sobre el nivel de un estudiante o sobre sus comportamientos sociales.

Las evaluaciones psicopedagógicas se deben recoger:

- Aspectos biológicos
- Aspectos intelectuales
- Aspectos sociales y emocionales
- Competencia curricular en las diferentes áreas
- Análisis del contexto escolar y sociofamiliar

La evaluación de las competencias curriculares son llevadas por distintos profesionales a lo largo del proceso. La primera responsabilidad es del profesor del aula, en colaboración con el profesor de apoyo y posteriormente psicólogos o pedagogos.

Las características de la evaluación del nee son:

- Dinámica: al valorar a un sujeto y su ambiente, hay que utilizar una metodología con enfoque cualitativo antes que cuantitativo, utilizando la descripción de situaciones sujetas al cambio.
- Ecológica: Se tienen en cuenta las situaciones e iteraciones en las que se desenvuelven las necesidades educativas.
- Curricular: Hay una justificación en la adecuación de la respuesta educativa a unas necesidades determinadas.
- Heurística: generadora de hipótesis explicativas que faciliten la intervención
- Funcional: orientada a tomas de decisiones sobre el proceso de enseñanza y aprendizaje.
- Colaborativa: intervienen varios profesionales con distintas responsabilidades, en un plano de igualdad, no jerárquico.
- Idiosincrásica: centrada en la realidad de cada persona

Es importante para realizar una evaluación conocer las deficiencias que presenta cada alumno, así como el análisis previo del ambiente del aula. Los alumnos con necesidad educativa especial pueden presentar diversos grados, sean por dificultad de aprendizaje, dificultad de comunicación, problemas físicos-sensoriales, o por el exceso de aprendizaje del alumno, dado que el alumno capta y aprende demasiado rápido, por lo que necesita estar avanzando y aprendiendo más cosas que los demás. (superdotado)

Para todos los casos, deben realizarse adaptaciones curriculares y buscar la metodología o estrategia de trabajo adecuada para poder satisfacer aquellas necesidades educativas especiales o para el último caso, realizar una aceleración para compensar su extra de conocimiento en otro curso superior.

#### Recursos internos para la detección y evaluación del NEE

Los recursos humanos para garantizar la detección de alteraciones del desarrollo de alumnos con necesidades educativas especiales:

- Director del centro: su tarea principal es apoyar el trabajo que el resto de profesores realiza con estos alumnos, convoca reuniones para el estudio de estos casos, garantiza y coordina la formación permanente del equipo docente, informar de los padres de la evolución y tratamiento que se lleva con sus hijos, así como coordinar los recursos internos con los grupos externos profesionales.

- Maestro en Pedagogía Terapéutica: Trabaja con niños especiales en aulas de apoyo. Entre sus funciones son: realiza el diagnóstico diferencial de las dificultades de aprendizaje, elaboración de recursos didácticos y programaciones para aquellos alumnos que lo necesiten, realiza la evaluación continua de los alumnos con NEE, asesora y orienta a los familiares de estos alumnos.
- Maestro especialista en audición y lenguaje: Profesional que detecta los problemas de comunicación y facilita las estrategias de intervención, elabora material para el profesorado.
- Profesor de apoyo: Profesional especialista en la compensación de desventajas sociales. Trabaja con minorías étnicas y culturales.
- Maestro especialista en educación especial: es el principal responsable de desarrollar las funciones de refuerzo pedagógicas, en los centros ordinarios que escolaricen a alumnos con NEE.
- Fisioterapeutas: Tratar alumnos con NEE de tipo motor, orienta a los profesores sobre las posturas más convenientes así como a la mejor forma de potenciar sus capacidades.
- ATE: Los profesionales para aquellos alumnos en situación de ausencia grave de autonomía personal que requieren atención específica de acompañamiento y ayuda para el desarrollo de hábitos de autonomía personal.

Para la consecución de este proyecto vamos a fijarnos más en las aptitudes necesarios para el seguimiento por parte de los ATE. Recordemos, como se indica en la descripción de esta documentación, el objetivo fundamental es la de proporcionar un sistema centralizado donde los ATE puedan capturar la información necesario para realizar sus diagnósticos.

## Motivación

Como dedicación e interés personal, interesa afrontar un reto importante para cualquier profesional del sector educativo, es la detección de las necesidades educativas especiales, que se presentan de cualquier forma en el entorno educativo. La detección preventiva es muy importante ya que afecta positivamente al rendimiento escolar de alumno o alumna en cuestión. Su posterior análisis y tratamiento beneficia enormemente a los alumnos.

El principal objetivo de la aplicación es seguir el comportamiento de cada alumno, centralizar todas las mediciones y proporcionar herramientas de presentación como la de los informes, ya que de nada sirve un buen diagnóstico sin un buen tratamiento.

Mi propósito con esta herramienta es intentar mejorar la calidad de la educación de atención primaria, ayudando al profesorado en las tareas más complejas como son el

análisis del nivel de aprendizaje, las mediciones que se realizan sobre diversas necesidades, las observaciones mostradas por parte del alumnado, y la presentación de las conclusiones finales en documentos oficiales.

# **CAPÍTULO 2: PLANIFICACIÓN DEL PROYECTO**



## Introducción

Este producto está enfocado para el uso de los profesionales educativos, desde profesores o maestros de cualquier especialidad hasta los directivos de los centros.

Diseñado preferiblemente para cursos infantiles o de primaria. Cualquier alumno matriculado perteneciente al curso puede ser seguido a través de la aplicación.

Para el objetivo final del proyecto no se podrán implementar todas las funcionalidades para todos los profesionales, a cambio se implementarán aquellas relacionadas con el profesorado principal y los ATE, ambos cercanos e interesados en el seguimiento de los alumnos.

La aplicación tiene que ayudar a mejorar las expectativas educativas de los alumnos de cada curso. Toda la gestión se realiza a través de la aplicación, mejorando la objetividad de la detección de las necesidades especiales.

## Descripción

El sistema tiene como objetivo de almacenar los datos recogidos por los especialistas para el seguimiento del rendimiento de los alumnos.

Los datos recogidos servirán para aplicar una serie de estímulos a aquellos alumnos con problemas en realizar sus actividades.

Posteriormente se realiza el seguimiento viendo en cada registro, que alumnos de un determinado grupo tienen problemas con ciertas actividades a través de las medidas recogidas.

El software debe permitir automatizar la recogida de los datos pertinentes por el ATE y almacenarlos en cualquier medio físico para su posterior tratamiento.

## Tecnología Empleada

El proyecto está diseñado para trabajar con los últimos dispositivos móviles del mercado. Esto implica que por la trascendencia sean dispositivos Android los utilizados para desempeñar todas las funcionalidades del sistema relacionado con el seguimiento.

Como es conocido, dentro del universo de Android existe bastante fragmentación entre sus distintas versiones. Esto es un problema, porque no podemos hacer frente a una compatibilidad global/universal.

Por tanto compensa compatibilizar la aplicación desde la versión 2.2 (Froyo) hasta la última 4.x (KitKat) dado que la API que se utiliza no difiere.

Para el servidor web se utilizará Apache 2.4 bajo el lenguaje PHP. En principio no es necesaria ninguna configuración especial, aunque sería interesante aplicar un protocolo de seguridad HTTPS.

Para la base de datos, se utilizará MySQL 5.6 como saben software privativo totalmente gratuito y con gran soporte, en esta nueva entre los que destacan las siguientes mejoras:

Los aspectos destacables de MySQL 5.6 son:

- **Mejores tiempos de ejecución y diagnósticos de consulta** mediante una mejor capacidad de diagnóstico MySQL.
- **Optimizaciones de subconsultas:** simplifica el desarrollo de consultas al optimizar dichas consultas antes de la ejecución.
- **Mejores diagnósticos del optimizador**
- **Mejor escalabilidad y alta disponibilidad:**
- **Clúster de replicación de auto curación:** la incorporación de Identificadores y Utilitarios de Transacciones Globales simplifica la detección y recuperación automáticas ante fallas.
- **Clúster de replicación de alto desempeño**
- **Replicación de tiempo diferido**

## Objetivos

Para la realización de este proyecto, se tomarán los siguientes objetivos como pautas o hitos a implementar:

### Para los ATE:

- Seguimiento de los alumnos por categorías
- Inserción de datos y parámetros adjuntos para cada actividad
- Inserción de observaciones durante las mediciones de cada actividad
- Inserción de factores/estímulos.

Para posteriores Trabajos de Fin de Grado, se puede perfectamente ampliar las funcionalidades a los siguientes roles:

### Para el Maestro en Pedagogía Terapéutica:

- Realizar el seguimiento del aprendizaje de cada alumno en el curso actual
- Establecer observaciones grupales o individuales para el entorno educativos
- Establecer observaciones para los familiares para asesorarles sobre el comportamiento de su hijo/s o hija/s.
- Análisis de la información y creación de gráficos de seguimiento del rendimiento de cada alumno.
- Adjuntar recursos pedagógicos a cada seguimiento de los alumnos.

- Obtener informes de otros profesionales sobre el curso determinado, en tiempo real.
- Elaborar horarios de seguimiento especializados para cada alumno.

Para el director:

- Tener disponible cualquier informe de todos los cursos del centro

## Identificación de los Stakeholder

En este apartado explicaremos cuáles son los principales stakeholders que formarán parte del desarrollo del proyecto. Los cuales pueden ayudar a comprender ciertas áreas sobre las que trabajan.

### *2.1 Director del centro*

Se encarga de la gestión total de cada uno de los departamentos del centro educativo. Así como, el único representante del centro en las instituciones superiores. Sus funciones básicas son: elaborar informes sobre el rendimiento del centro, llevar el control financiero del centro.

### *2.2 Jefe de Estudios*

Es el representante de llevar a cabo las decisiones tomadas por el Consejo Escolar. Coordinar las actividades de carácter académico y velar su ejecución. Coordinador de la acción de los tutores y del orientador escolar.

### *2.3 Profesorado*

Trabajan a diario con los alumnos en múltiples disciplinas. Realizan informes para evaluar el rendimiento de cada alumno y global del curso. También realiza reuniones con los padres de cada alumno.

### *2.4 ATE*

Trabajan a diario con los alumnos con necesidades especiales. Realizan informes para evaluar el comportamiento de cada alumno especial y le ayudan a realizar cualquier tarea complicada. Se encarga de notificar al profesor cualquier incidencia u observación sobre un alumno en particular.

## Modelo de proceso para el desarrollo

Para el desarrollo de la aplicación utilizaremos la metodología en cascada dado que el software está prácticamente libre de cualquier requisito inestable que a corto plazo pueda cambiar la funcionalidad del sistema. Además esta metodología es ideal para un equipo mínimo de desarrolladores y la cantidad de información generada en cada etapa es más elevada que cualquier otra alternativa. El proyecto que nos ocupa tampoco es demasiado extenso ni complejo.

Gracias a la metodología en cascada presenta algunas variantes de la original, algunas de ellas permiten la retroalimentación entre algunas fases, siendo interesante para las etapas de análisis y diseño porque siempre hay algún que otro error que a corto plazo pueda extender los siguientes periodos de tiempo.

Como breve resumen del concepto de este proceso de desarrollo en ingeniería de software, se detallan las fases que se realizarán en este proceso:

### Etapa 1: Comunicación

Esta fase consiste en asentar las primeras conversaciones para constituir las bases del proyecto. Se recaban los primeros requerimientos, aquellas descripciones sobre qué consistirán los objetivos, fundamentales para la construcción del nuevo sistema. También se proporcionan información acerca de los stakeholders involucrados, como es el entorno organizativo donde se ejecutará el software, etc.

### Etapa 2: Planeación

En esta fase se cometen las primeras estimaciones tanto económicas/temporales como técnicas. Se elabora un calendario para planificar todo el proceso de desarrollo y también las políticas de seguimiento para todos los hitos del desarrollo.

### Fase 3: Fase de Modelado

Tanto el análisis como el diseño en un proceso software son muy importantes. Ambas fases constituyen la columna vertebral de todo el proceso, fundamental para una construcción estable.

En el análisis se utilizan una serie de técnicas las cuales el cliente plantea las necesidades que se presenta e intenta explicar lo que debería hacer el software o producto final para satisfacer dicha necesidad mientras que el desarrollador actúa como interrogador, como la persona que resuelve problemas. Con este análisis, el ingeniero de sistemas puede elegir la función que debe realizar el software y establecer o indicar cuál es la interfaz más adecuada para el mismo.

En el diseño fundamentalmente se caracteriza la solución a implementar en la plataforma correspondiente. Se definen la arquitectura, componentes, interfaces y otras características de un sistema o componente que sirvan de base para la construcción.

#### Etapa 4: Fase de Construcción

Esta es la fase de codificación e implementación de la solución proporcionada en el diseño con algún lenguaje de programación informática para una plataforma en concreto. Posteriormente las unidades ejecutables se van probando y analizando para comprobar que no existan errores funcionales en la ejecución normal del mismo.

#### Etapa 5: Fase de Despliegue

Esta fase consiste en entregar el producto al usuario y realizar la instalación en el entorno final del todo el sistema. También se abre un periodo de asistencia, donde entre otras tareas se enseña a los usuarios finales a manejar el nuevo software.

Dado que también se va a desarrollar la aplicación en Android, hay que trabajar mediante la metodología orientada a objetos. La metodología orientada a objetos consiste en usar los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

Los objetos son entidades que tienen un determinado *estado*, *comportamiento* (*método*) e *identidad*:

- El *estado* está compuesto de datos o informaciones; serán uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- El *comportamiento* está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, qué operaciones se pueden realizar con él.
- La *identidad* es una propiedad de un objeto que lo diferencia del resto; dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).

Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento.

Los **métodos (comportamiento)** y **atributos (estado)** están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a alguno de ellos.

## Entorno Operativo y Arquitectura

El sistema consistirá en una aplicación móvil que permita gestionar la información necesaria para llevar la gestión de las necesidades de cada alumno o del grupo en general. Para ello utilizaremos como plataforma Android (sus características serán explicadas en apartados siguientes) dado que es un sistema versátil, y con una cuota de mercado enorme con respecto a las otras alternativas. Además existe una buena comunidad de soporte para el desarrollo lo que nos facilita buenas prácticas, solución a errores, y otra serie de procedimientos para garantizar una buena calidad del producto final.

Así mismo, el sistema debe comunicarse con un servidor externo conectado a una base de datos, para extraer o almacenar la información de la aplicación móvil.

Con la implementación de un servidor web es posible “portar” este sistema a una aplicación web, donde cualquiera que no tenga un dispositivo Android, pueda utilizar el mismo sistema en sus ordenadores personales.

## Arquitectura Física

El sistema actual está basado en un modelo cliente-servidor de dos capas. Por un lado, los clientes son los dispositivos móviles Android que se conectarán a través de una aplicación al servidor. El servidor tramita todas las peticiones con ayuda de un sistema gestor de base de datos externo.

La base de datos será relacional y estará basada en MySQL.

Como sugerencia para hacer más robusto el sistema, se puede implementar balanceadores de carga entre cada nodo de servidores. Incluso añadir un segundo nodo de datos para realizar réplicas entre las bases de datos.

En el apartado de seguridad, incluir varios firewall: el primero entre la aplicación móvil y el servidor, y el segundo, entre el nodo servidor y la base de datos. Esta decisión optimizará la escasa probabilidad de ataques con éxito en el servidor, y en el peor de los casos, se garantiza por segunda vez un ataque a la base de datos, muy importante de evitar ya que se almacena información personal acerca de los usuarios.



*Ilustración 3- Arquitectura Física*

### Arquitectura Lógica

La arquitectura lógica está basada en un modelo de dos capas cliente-servidor. El lado cliente será una aplicación móvil escrita en Java a través del Google API de Android. Se va a utilizar la versión 4.3 para compatibilizar también la aplicación con las tabletas y/o últimos smartphone.

El servidor construido bajo Apache servirá la información necesaria al cliente mediante objetos JSON. De esta manera, en un futuro se podría agregar un servidor web para utilizar también el servicio mediante intranet ampliando la escalabilidad del sistema.

Los datos serán recogidos mediante el sistema gestor de base de datos por excelencia **MySQL**, gratuito y con gran soporte, además de robusto para el número de conexiones que se ha estudiado. Incluso se puede implementar un pool que nos garantice el establecimiento de las conexiones, al mismo tiempo, reduce la cantidad de memoria por cada petición.

Después del resumen de como sería la idea principal para el diseño, comenzamos a detallar el paradigma utilizado.

En esta ocasión se utilizara también el modelo vista- controlador inherente cuando se desarrolla con sistemas Android. El paso de mensajes o de datos en este caso se realizara a través de la tecnológica JSON entre la capa cliente y la capa servidor. Para realizar esto es necesario usar una clase auxiliar para manejar los tipos JSON y realizar conexión con el servidor.

[Imagen]

Es muy importante recordar que para el desarrollo de este proyecto se ha empleado la máquina virtual del Android SDK, por lo que en tema de conexiones de red se refiere, la IP puede no ser la misma que la de una intranet.

### Android

Es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto la fundación del Open Handset

Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.



*Ilustración 4- Detalle sobre arquitectura Android*

Para empezar con el desarrollo de aplicaciones en Android es importante conocer cómo está estructurado este sistema operativo. En el caso de Android está formada por varias capas que facilitan al desarrollador la creación de aplicaciones. Además, esta distribución permite acceder a las capas más bajas mediante el uso de librerías para que así el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware de los teléfonos.

Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como *pila*.

Para explicar mejor cada una de las capas, nos apoyamos en el siguiente diagrama, comenzando de abajo hacia arriba:



*Ilustración 5- Detalle extendido sobre arquitectura lógica de Android*



- **Kernel de Linux.** El núcleo del sistema operativo Android está basado en el kernel de Linux versión 3.x , similar al que puede incluir cualquier distribución de Linux, como Ubuntu, pero adaptado a las características del hardware de dispositivos móviles.

El núcleo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura. El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores. De esta forma también nos evitamos problemas con la compatibilidad de ciertas características precisas de cada teléfono. Si necesitamos hacer uso de la cámara, el sistema operativo se encarga de utilizar la que incluya el teléfono, sea cual sea. Para cada elemento de hardware del teléfono existe un controlador (o *driver*) dentro del kernel que permite utilizarlo desde el software.

El kernel también se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, etc.) y del sistema operativo en sí: procesos, elementos de comunicación (*networking*), etc.

- **Librerías.** La siguiente capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Estas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma “más eficiente”.

Entre las librerías incluidas habitualmente encontramos OpenGL (motor gráfico), Bibliotecas multimedia (formatos de audio, imagen y video), Webkit (navegador), SSL (cifrado de comunicaciones), SQLite (base de datos), entre otras.

- **Entorno de ejecución.** Como podemos apreciar en el diagrama, el entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí encontramos las librerías con las funcionalidades habituales de Java así como otras específicas de Android.

El componente principal del entorno de ejecución de Android es la máquina virtual **Dalvik**. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una única vez y de esta forma estarán listas para distribuirse con la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación.

Cabe aclarar que Dalvik es una variación de la máquina virtual de Java, por lo que **no es compatible con el bytecode Java**. Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión .dex que es específico para Dalvik, y por ello no podemos correr aplicaciones Java en Android ni viceversa.

- **Framework de aplicaciones.** La siguiente capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik.

Siguiendo el diagrama encontramos:

- **Activity Manager:** Se encarga de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.
- **Windows Manager:** Se encarga de organizar lo que se mostrará en pantalla. Básicamente crea las superficies en la pantalla que posteriormente pasarán a ser ocupadas por las actividades.
- **Content Provider:** Esta librería es muy interesante porque crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
- **Views:** En Android, las vistas los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.
- **Notification Manager.** Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Un dato importante es que esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.
- **Package Manager:** Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con paquete nos referimos a la forma en que se distribuyen las aplicaciones Android, estos contienen el archivo .apk, que a su vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.
- **Telephony Manager:** Con esta librería podremos realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
- **Resource Manager:** Con esta librería podremos gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es

decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o layouts. En un post relacionado a la estructura de un proyecto Android veremos esto más a fondo.

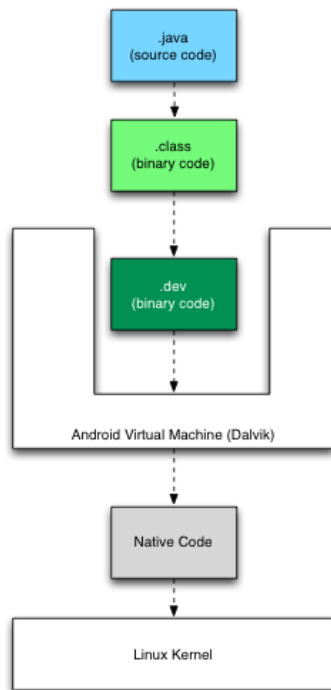
- **Location Manager:** Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.
  
- **Sensor Manager.** Nos permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
  
- **Cámara:** Con esta librería podemos hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.
  
- **Multimedia:** Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

**Aplicaciones.** En la última capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.

En esta capa encontramos también la aplicación principal del sistema: Inicio (Home) o lanzador (launcher), porque es la que permite ejecutar otras aplicaciones mediante una lista y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso widgets, que son también un tipo de aplicaciones de esta capa.

#### Android Virtual Machine (Dalvik)

Android soporta una máquina virtual diferente denominada **Dalvik**. Android y Dalvik son los encargados de trabajar con el código fuente Java y convertirlo en código nativo para el kernel de linux que la plataforma Android soporta.

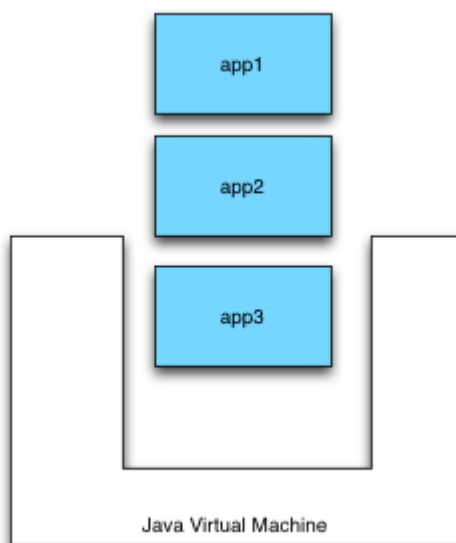


*Ilustración 6- Ejemplo sobre ficheros .class en Android*

Como la imagen muestra Android añade un paso intermedio. En este paso todos los **ficheros .class de Java son pasados por otro proceso adicional de optimización** ya que recordemos estamos en una plataforma móvil y los recursos son limitados. En este proceso todos los ficheros .class son agrupados en un único fichero **.dex (Dalvik Executable)**.

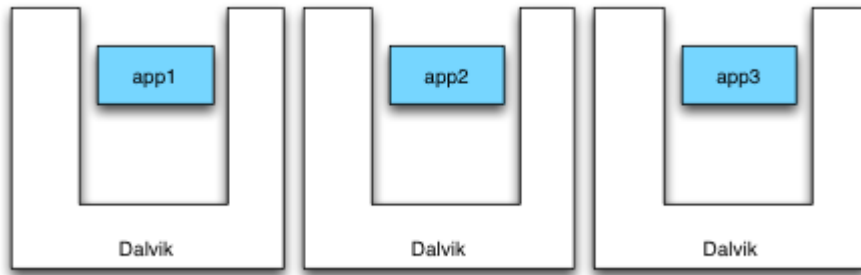
#### Dalvik y Multi JVM

Otra diferencia importante que existe entre las aplicaciones Java y las aplicaciones Java Android (Dalvik) es que por ejemplo en un servidor de aplicaciones (Jboss, WebSphere, WebLogic) todas las aplicaciones corren sobre la misma máquina virtual.



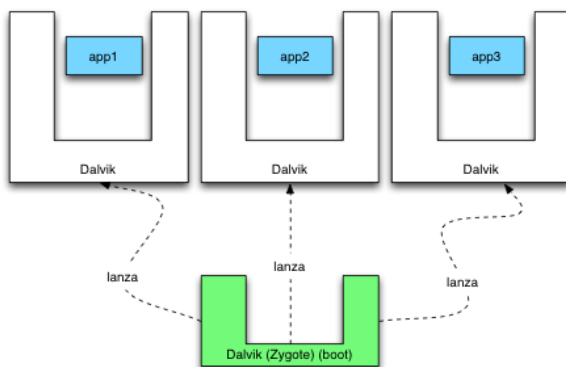
*Ilustración 7- Ejecución de varias aplicaciones en una máquina virtual normal*

**En Dalvik esto no es así** ya que cada aplicación corre en su propia máquina virtual aislada con el aumento de recursos que esto supone pero también con el mayor aislamiento entre las distintas aplicaciones que se consigue.



*Ilustración 8- Ejecución de varias aplicaciones en Dalvik*

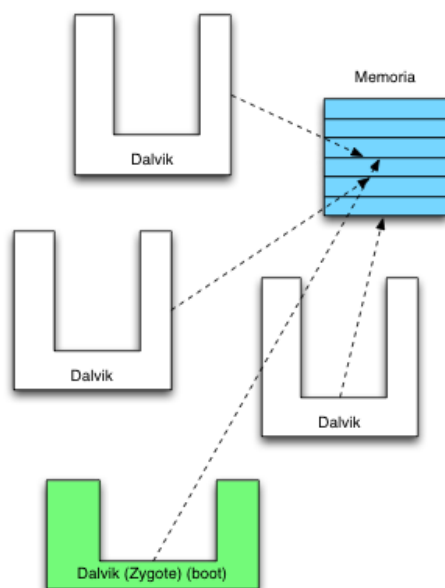
Aunque disponemos de una máquina virtual por cada aplicación Android realiza una fuerte optimización y carga una máquina virtual inicial denominada Zygote que es la encargada de arrancar el resto.



*Ilustración 9- Dispatcher en Android*

Dalvik Memoria

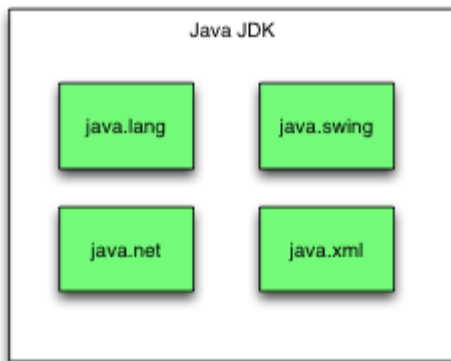
Todas las máquinas virtuales comparten la memoria del terminal



*Ilustración 10- Pila de memoria en Android*

Android y Java SDK.

El JDK de Java almacena sus clases principalmente en el **fichero rt.jar** . Este fichero contiene todas sus clases con todos los paquetes.

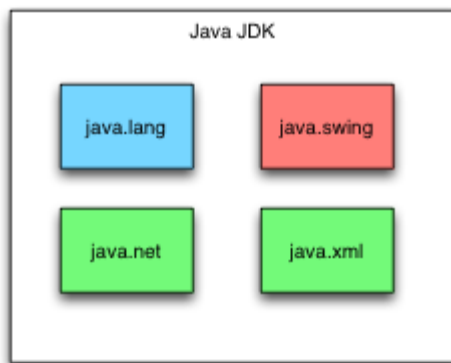


*Ilustración 11- Diagrama de paquetes*

Lamentablemente en el Android SDK esto no es así ya que Android no necesita tantas clases como el JDK standard para trabajar. De hecho hay paquetes que sobran por completo (caso de Swing) . Concretamente en Android podremos dividir los paquetes del JDK de la siguiente forma.

- Paquetes implementados completamente (Verde)
- Paquetes implementados parcialmente (Azul)
- Paquetes no implementados (Rojo)

Con lo cual el gráfico quedaría de la siguiente forma



*Ilustración 12- Diagrama de paquetes*

## Herramientas Empleadas

Durante el tiempo de desarrollo del proyecto se ha tenido que recurrir al uso de numerosas herramientas con las que poder materializar la idea planteada.

Para estructurar este apartado, se ha optado por agruparlas en función de la etapa en la que haya sido utilizada. A continuación, se detalla una breve descripción de cada una de ellas:

### Hardware

1. Ordenador portátil: HP PAVILION DV5
  - Intel® Core Duo a 2 GHz
  - 4 GB RAM
  - 320GB HDD
2. Ordenador sobremesa.
  - AMD FX-8350 – 8 núcleos a 4 GHz
  - 8 GB RAM
  - 2TB HDD
3. Smartphone: LG G2.
  - Procesador Snapdragon 4 núcleos a 2,3 GHz y Gpu Adreno 330
  - 2 GB RAM
  - 16GB memoria interna

### Software

- Microsoft® Windows 8.1: Sistema operativo del ordenador utilizado para el desarrollo del proyecto.
- Microsoft® Office 2010: Paquete de aplicaciones de ofimática. Las principales herramientas utilizadas de este paquete han sido:
  - Microsoft® Word: Procesador de textos, para la realización de la memoria y distintos documentos auxiliares.
  - Microsoft® Power Point: Potente herramienta para realizar presentaciones.
- OpenProj: Programa gratuito que permite la realización de la planificación y seguimiento del proyecto. Tras la realización de la planificación, y generación del diagrama Gantt, esta herramienta permite registrar el avance del proyecto y vigilar sus desviaciones, por tanto afectaría tanto a la fase de planificación, en un primer momento como al resto de etapas del proyecto para vigilar su avance.

### **Etapa de análisis y diseño**

- WhiteStarUML: Programa gratuito que permite diseñar todo tipo de diagramas UML (secuencia, colaboración, clases). Además permite exportar éstos a formato .jpg para su visualización en procesador de textos.

### **Etapa de Implementación**

- Eclipse Kepler: Ha sido la herramienta empleada en la etapa de implementación. Además para facilitar la programación de la aplicación del Smartphone, ha sido necesario incorporar el Android

SDK (Software Development Kit) al Eclipse. Por tanto el lenguaje de desarrollo que se ha utilizado en este entorno es el lenguaje orientado a objetos Java, con las propiedades añadidas de Android.

- SQLyog: Esta herramienta, ha permitido diseñar, comprobar y consultar la base de datos basada en MySQL.



## Estudio de Viabilidad

En este apartado se explicará y detallará brevemente aquellos aspectos utilizados para realizar el estudio de viabilidad del proyecto. El proyecto sí es viable de acuerdo a los diferentes parámetros que se estudian a continuación.

### Viabilidad Tecnológica

#### *Viabilidad Técnica - Hardware*

Atendiendo a los requerimientos hardware que puedan surgir durante el desarrollo del proyecto, serán necesarios ordenadores para realizar la implementación, un servidor y una base de datos implementado de forma local

En cuanto a las necesidades que puedan surgir una vez que el sistema sea puesto en funcionamiento, se requerirá de una arquitectura física basada en cliente-servidor de N capas que garantice la disponibilidad del sistema en caso de que haya muchos usuarios conectados al mismo tiempo. Para ello, podemos disponer de dos balanceadores de carga, uno activo y otro pasivo (en caso de fallo), y un servidor web (opcional) ya que en un futuro pueden necesitar la Web para utilizar la aplicación desde los ordenadores.

En cuanto a la seguridad, que también es uno de los objetivos vitales del proyecto, se accederá al sistema a través de un firewall.

También es necesario un servidor de datos centralizado, conectado a los servidores, para así disponer de replicación de datos por red, básico para aumentar la robustez de los recursos que contiene nuestra aplicación.

Como siempre, incorporar seguridad a un sistema tiene un coste asociado elevado pero para este tipo de aplicaciones en entorno real es imprescindible.

Estos componentes no se añaden al presupuesto de este proyecto.

#### *Viabilidad Técnica - Software*

El sistema operativo con el que se desarrollará el proyecto será Windows, aunque durante el funcionamiento deberá poder ser utilizado con otros sistemas operativos como Linux o Mac, e incluso desde cualquier sistema operativo móvil Android.

Como entorno de desarrollo integrado, IDE, se utilizará Eclipse junto con todo lo necesario para el desarrollo de Android.

Además, según estudios realizados sobre el entorno de ejecución los actuales centros educativos tienen subsistemas independientes gestionados por la Junta, el sistema deberá interoperar con:

- Sistema de gestión del alumnado.
- Sistema de gestión del personal.

Algunos de estos sistemas podrían estar basados en servicios LDAP ya implementados en las organizaciones escolares.

Pueden surgir nuevas actualizaciones para mejorar pequeños errores o incidencias, o implementar alguna otra funcionalidad. A priori, al tratarse de una arquitectura cliente-servidor, los detalles de implementación y diseño de las actualizaciones así como su despliegue quedarán supervisados más adelante conforme se tomen las políticas oportunas durante el desarrollo del proyecto.

El impacto de las actualizaciones para el cliente será bajo, puesto que los datos no se modificarán.

He tratado de utilizar todos los componentes open-source para abaratar lo máximo posible los costes.

### *Viabilidad Temporal*

En este punto se analizará la distribución del factor tiempo durante la planificación de desarrollo de la aplicación.

Dada la naturaleza del proyecto, el periodo de Trabajo Fin de Grado tiene un límite de desarrollo de entre 5 y 8 meses máximos (contando desde Febrero del año anterior). No existe margen de ampliación.

Teniendo en cuenta que la jornada laboral es de 8 horas al día, trabajando 22 días al mes durante los 8 meses máximo que durará el proyecto, se obtiene un total de 1480 horas de trabajo.

Según proyectos anteriores, los porcentajes estimados que se dedicarán a cada etapa son:

	<b>Tareas</b>	<b>%</b>	<b>Horas estimadas</b>
Estudio de Viabilidad	Planificación	5%	70,4
Fase de Desarrollo	Requisitos del sistema	10%	140,8
	Análisis de componentes	20%	281,6
	Diseño de componentes	15%	211,2
	Implementación de componentes	40%	563,2
	Pruebas de componentes	10%	140,8
Fin	Documentación	5%	70,4
<b>Total</b>		<b>100%</b>	<b>1480</b>

*Tabla 1- Porcentajes sobre la planificación del proyecto*

Tras analizar todos estos datos, se estima que la duración total del proyecto será de 65 días naturales, es decir, aproximadamente dos meses y una semana.

A continuación, se muestra un diagrama de Gantt, que permite ver, de una forma gráfica y clara, las tareas a realizar a lo largo del tiempo en el periodo de duración del proyecto. Se puede observar las fechas a cumplir en cada una de las tareas y subtareas.

### *Viabilidad Financiera*

El presente apartado se centra en el estudio económico, con su estimación de costes mediante un estudio realizado de forma orientativa con el que poder hacer frente a los costes que se producen en el desarrollo del producto software.

Resumiremos el presupuesto en las siguientes tablas:

#### Parte 1- Estimación basada en Puntos de Función

La métrica del punto de función es un método en ingeniería del software para medir el tamaño del software. Pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

<b>Factores de complejidad</b>	<b>Rango [0-5]</b>	<b>Factores de Complejidad</b>	<b>Rangos [0-5]</b>
Comunicación de datos	4	Funciones distribuidas	3
Rendimiento	4	Configuraciones fuertemente utilizadas	5
Frecuencia de transacciones	4	Entrada online de datos	5
Diseño para la eficiencia del usuario final	3	Actualizaciones online	1
Procesos complejos	3	Reusabilidad	3
Facilidad de instalación	5	Facilidad de operación	4
Instalaciones en múltiples lugares	4	Facilidad de cambio	4

<b>Parámetro</b>	<b>Complejidad baja</b>	<b>Complejidad Media</b>	<b>Complejidad Alta</b>
Entradas	X4	X2	X2
Salidas	X3	X3	-

Ficheros internos	-	-	-
Ficheros externos	X2	-	-
Consultas externas	-	-	-

Para hallar los puntos de función, debemos seguir una serie de pasos:

1.- Identificar el número de elementos y complejidad:

- Entradas:

- Información sobre alumnos
- Información sobre cursos
- Información sobre ATE
- Información sobre actividades
- Información sobre observaciones
- Información sobre mediciones
- Información sobre factores
- Información sobre estímulos

- Salidas:

- Listado de las actividades por tiempo
- Listado de las actividades por estímulos
- Listado de las actividades por factores
- Listado de alumnos
- Listado de mediciones
- Listado de observaciones

- Ficheros externos:

- Base de datos del profesorado
- Base de datos de los alumnos

El siguiente paso es identificar los puntos de función sin ajustar (PFNA):

- Total de entradas = 8
- Total de salidas = 6
- Total de ficheros externos = 2
- Total de ficheros internos = 0
- Total de cuestiones externas = 0

Existen ocho entradas, cuyas complejidades son las que se explican a continuación. La inserción de datos relativos a cada alumno tiene una complejidad media dado que cada alumno tiene un perfil distinto. Los datos relacionados con el profesorado tienen una complejidad baja ya que solo almacena datos identificativos. La inserción de datos sobre los cursos tiene una complejidad baja. La inserción de las actividades tiene una complejidad media ya que solo hay que identificarle mediante un nombre y asociarle un curso y un alumno o grupo de alumnos. Las inserciones de las observaciones son de complejidad alta por su distinta naturaleza y el sistema de igual manera tiene que interpretarlo correctamente. La inserción de mediciones tiene una complejidad alta dado que pueden ser de distinta naturaleza con información muy

variada y extensa. Las inserciones de estímulos y factores al requerir pocos campos de entrada tendrán una complejidad baja.

Con respecto a las 6 salidas, en primer lugar el timeline (o los diferentes listados) tiene una complejidad alta, ya que las divisiones pueden ser temporales o por factores determinados por el propio educador. Los diferentes tipos de listados tendrán una complejidad media puesto que solo mostrarán sus datos correspondientes.

El fichero externo relativo a los datos del alumnado y profesorado tendrán una complejidad alta, ya que también se incluye toda la historia de cada alumno por cada uno de los cursos.

Con estos datos medidos y analizados, procedemos a calcular la cuantía del PFNA para realizar las estimaciones de costes oportunas:

**PFNA** = (2 entradas medias x4+ 2 entradas altas x6+ 4 entradas bajas x3)+ (3 salidas altas x7+ 3 salidas medias x5) + (2 ficheros externos altos x10) = 88

Ahora calculamos el factor de ajuste, siendo  $\Sigma F_i$  la suma de los factores de complejidad:

$$\mathbf{FA} = (0'01 \times \Sigma F_i) + 0'65 = (0'01 \times 35) + 0'65 = 1$$

Por último, calcularemos los puntos de función ajustados usando los PFNA y el FA de la siguiente manera:

$$\mathbf{PF} = \Sigma F_i \times \mathbf{PFNA} = 1 \times 88 = 88$$

## Parte 2- Estimación usando COCOMO:

El Modelo Constructivo de Costes (**o COCOMO**, por su acrónimo del inglés **CO**nstructive **CO**st **MO**del) es un modelo matemático de base empírica utilizado para estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

Podremos usar uno de los tres siguientes modelos:

- **Modelo orgánico**: un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar.

El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).

- **Modelo semilibre:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modelo rígido:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

La naturaleza de este proyecto presenta fuertes restricciones técnicas en cuanto a la funcionalidad y algunas tecnológicas. Además la experiencia con un sistema Android es prácticamente escasa y la aplicación debe sincronizarse perfectamente con los servidores. Por ello se utiliza el modo rígido.

MODO	a	b	c	d
<b>Orgánico</b>	2`4	1`05	2`5	0`38
<b>Semilibre</b>	3	1`12	2`5	0`35
<b>Rígido</b>	3`6	1`2	2`5	0`32

Tabla 2- Modelos Cocomo

La tabla de líneas de código usando un lenguaje se muestra también a continuación:

Lenguaje	PF
Ensamblador	320
C	150
Cobol	150
Pascal	91
Basic	64
PCL	64
Java	53
C++	29

Tabla 3: Estimaciones reales cedidas por [www.ttn.gov.ar](http://www.ttn.gov.ar)

Sabiendo que los puntos de función ajustados nos han dado el resultado de 88 y que el lenguaje que se utilizará será Java, debido a que es un lenguaje con el que se puede interactuar a un nivel alto, mediante una máquina virtual, hallaremos el número de líneas de código del programa:

$$LDC \text{ Totales} = \text{Líneas de código del lenguaje} \times FA = 53 \times 88 = 4664$$

$$LDC \text{ Totales} \approx 4,664 \text{ KLDC}$$

<i>FACTORES/ RANGOS</i>	<i>Muy Bajo</i>	<i>Bajo</i>	<i>Medio</i>	<i>Alto</i>	<i>Muy Alto</i>
<b>Fiabilidad Requerida</b>	-	-	-	<b>1,15</b>	-
<b>Tamaño de la BD</b>	-	-	-	-	<b>1,16</b>
<b>Complejidad del SW</b>	-	-	-	<b>1,15</b>	-
<b>Restricciones de tiempo de ejecución</b>	-	-	-	<b>1,11</b>	-
<b>Restricciones de memoria</b>	-	-	-	<b>1,06</b>	-
<b>Volatilidad de HW</b>	-	<b>1</b>	-	-	-
<b>Restricciones de tiempo de respuesta</b>	-	-	-	<b>1,07</b>	-
<b>Calidad de los analistas</b>	-	-	<b>1</b>	-	-
<b>Experiencia con el tipo de aplicación</b>	-	<b>1,13</b>	-	-	-
<b>Experiencia con el HW</b>	-	-	<b>1</b>	-	-
<b>Experiencia con el lenguaje de programación</b>	-	-	<b>1</b>	-	-
<b>Calidad de los programadores</b>	-	-	-	<b>0,86</b>	-
<b>Técnicas modernas de programación</b>	-	-	-	<b>0,91</b>	-
<b>Empleo de herramientas</b>	-	-	<b>1</b>	-	-
<b>Restricciones a la duración del proyecto</b>	-	-	-	<b>1,04</b>	-

Se han elegido estos valores ya que se necesita una alta fiabilidad requerida de la aplicación para que se vea que funcione de forma correcta, el tamaño de la base de datos es muy alta teniendo en cuenta los datos ya establecidos previamente y los nuevos que se insertarán. El software es de complejidad alta, ya que la aplicación debe interoperar con los servidores y sincronizarse con ellos. Las restricciones del tiempo de ejecución serán del valor alto al necesitar una rapidez de ejecución por parte de la aplicación.

Las restricciones de memoria tanto del dispositivo móvil es muy importante controlar y mantener un rendimiento altamente aceptable, y tampoco hay que descuidar la carga en términos de memoria del servidor para cada una de las conexiones.

La calidad de los analistas será media al tener una experiencia demostrada a lo largo de la titulación. La experiencia en este tipo de aplicaciones y arquitecturas el nivel es bajo así como su experiencia con el hardware en cuanto a la programación para dicho hardware. El nivel de experiencia con el lenguaje de programación será alto.

La calidad como programador es alta, al estar habituado a desarrollar software durante mucho tiempo y haberse incorporado al mundo laboral durante años. Sin embargo, las técnicas modernas de programación se pueden poner en práctica en el presente proyecto, como el paradigma orientado a objetos y alguno de sus patrones de diseño.

El empleo de herramientas será también de un nivel alto, al saber manejar las herramientas necesitadas y una comprensión de las librerías utilizadas para la realización del proyecto.

Por último, las restricciones a la duración del proyecto deben ser de un valor alto al presentarse el proyecto en el presente año.

Se muestra en negrita los datos que se cogen para calcular el peso del factor de ajuste, el cual es el siguiente.

$$m(x) = \prod m(x_j) = 1'15 \times 1'16 \times 1'30 \times 1'11 \times 1'21 \times 1 \times 1,07 \times 1 \times 1'13 \times 1'00 \times 1'00 \times 0'86 \times 0'91 \times 1 \times 1'04 = 2,29$$

$$m(x) = \mathbf{2,29}$$

Teniendo ya el peso del factor de coste  $x_j$  calculado, podemos calcular el esfuerzo realizado de la aplicación en personas/mes:

$$E = a * KLDC^b = 3'6 * 4,664^{(1'2)} = 22,84 \text{ personas/mes}$$

$$T = c * E^d = 2'5 * 22'84^{(0'32)} = 6'8 \text{ meses}$$

Hardware utilizado

HARDWARE	USO(%)	UNIDADES	COSTE(€)	COSTE TOTAL (€)
Servidor	100	1	1500	1500
Ordenador Personal	100	1	800	800
Conexión a Internet	100	1	35,95	35,95



*Tabla 4- Listado de componentes HW necesarios en el desarrollo*

<u>Software Utilizado</u>				
<b>SOFTWARE</b>	<b>USO(%)</b>	<b>UNIDADES</b>	<b>COSTE(€)</b>	<b>COSTE TOTAL (€)</b>
Windows 7 Professional	100	1	126€	630€
Adobe DreamWeaver CS6	100	1	300€	300€
Microsoft Office 2010	100%	1	100€	100€
MySQL Community Edition	100	1	0€	0€
XAMPP	100	1	0€	0€
phpMyAdmin	100	1	0€	0€
Eclipse IDE	100	1	0€	0€

*Tabla 5-Listado de componentes SW necesarios en el desarrollo*

<b>RECURSOS HUMANOS</b>	<b>TIEMPO (horas)</b>	<b>COSTE</b>
Ingeniero	1480	22€/hora

*Tabla 6- Costes de recursos humanos*

<b>TIPO</b>	<b>COSTE</b>
Hardware	2.335,95€
Software	3.129,95€
Recursos Humanos	32.560€

*Tabla 7 - Costes totales por categorías*

Conclusión: el presupuesto se encuentra dentro de los límites temporales y económicos proporcionados.

### **Metodología Empleada**

Toda metodología no es más que un conjunto de métodos, técnicas, herramientas y soporte documental que ayudan a cubrir más de una etapa del ciclo de vida. Ese conjunto de métodos, técnicas y herramientas deben especificar cómo se tiene que dividir el proyecto en las fases, etapas y actividades correspondientes; así como dar una explicación clara y práctica de cómo llevar a cabo cada tarea en concreto.

La metodología representa el camino para desarrollar el software de una forma sistemática. Esto se logra gracias al control y seguimiento del trabajo marcado por las salidas que se producen y por las restricciones aplicadas al comienzo del mismo. Para el

desarrollo del Software, también se deberá de tener en cuenta la documentación que se genere.

Son varias las razones de peso para optar por la Metodología Orientada a Objetos, para la realización de este Proyecto Software. En primer lugar, es la metodología que proporciona el Android SDK para la programación en Android. Además, al ser la ampliación y mejora de un proyecto ya existente, se ha optado por continuar usando la metodología del proyecto original, que también corresponde a una metodología Orientada a Objetos. Gracias a ella se pueden manejar sistemas con una alta complejidad y permite estructurar la arquitectura software en niveles y capas de tal manera que se permitiese un acceso flexible, versátil e independiente en cada una de ellas así como un mejor mantenimiento del programa.

Básicamente, esta metodología permitió escribir el software de tal forma, que el código quedase organizado de la misma manera que la aplicación original. De esta manera se permite aprovechar sus ventajas: la herencia, polimorfismo y redefinición de métodos para conseguir una mayor funcionalidad, manteniendo el código original,

Las ventajas que ofrece esta metodología son:

- **Compresión:** Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases. Estas clases se corresponden con las estructuras de información.

- **Uniformidad:** La representación de los objetos implica el análisis, el diseño y la codificación de los mismos.

- **Mantenimiento:** Permite realizar un tratamiento diferente entre aquellos objetos que permanecen constantes en el tiempo y los que cambian con frecuencia, se pueden aislar las partes del programa que permanecen inalterables en el tiempo.

- **Reusabilidad:** La noción de clase permite la reutilización de las definiciones de clases empleadas para otros programas e incluso de los procedimientos que las manipulan.

La metodología se ha de aplicar dentro de un entorno. Para ello, se necesita conocer, además del entorno mismo sobre el que se aplica la tecnología, los factores que influyen en la elección de la misma.

Las características que debe tener una buena metodología vienen marcadas por los siguientes puntos:

1. Debe cubrir el mayor número de etapas de desarrollo de un sistema de información.
2. Mostrar una cobertura total del ciclo de vida del Software.
3. Facilitar la comunicación entre aquellas personas que están involucradas en su desarrollo.
4. Tiene que hacer visible y controlable el avance del sistema que se desarrolla.
5. Facilitar la gestión y el seguimiento del proyecto.

6. La metodología debe contener actividades que mejoren el proceso de desarrollo.
7. Debe incluir la definición de las restricciones del sistema.
8. Por último, tiene que soportar la validación y verificación de toda la documentación generada.

Con todo esto, se puede decir, que si el ciclo de vida nos sirve para indicarnos que es lo que hay que obtener a lo largo del proyecto y los procesos que deben de estar presentes para que esto se pueda conseguir, la metodología será la encargada de indicar cómo se tiene que proceder con cada una de las actividades que forman el proceso.

# **CAPÍTULO 3: ANALISIS DEL SISTEMA**

Tras analizar toda la documentación disponible y realizar entrevistas, se generó un nuevo documento de especificación de requisitos cuyo contenido se resume en las tablas de los siguientes apartados.

**OBJETIVOS DEL SISTEMA**

Los objetivos del sistema son aquellas actividades que el sistema debe cumplir para proporcionar una correcta funcionalidad a los usuarios.

En nuestro caso, el sistema debe cumplir:

OBJ-1	Inserción, modificación y borrado de actividades
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe ser capaz de almacenar cada una de las actividades. Así como modificar su datos o borrarlos de cualquier tipo de almacenamiento.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	
OBJ-2	Seguimiento individual y grupal de los alumnos
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe realizar el seguimiento individual de cada alumno dependiendo de los factores establecidos previamente por el ATE. De la misma manera un seguimiento grupal por cada curso.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	
OBJ-3	Inserción y borrado de mediciones
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe ser capaz de almacenar e interpretar cada una de las mediciones de una actividad concreta. Cada medición se realiza sobre una día concreto del curso actual
Subobjetivos	Ninguno

Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	

<b>OBJ-4</b>	Inserción, modificación o borrado de observaciones
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe permitir la inserción, modificación de datos y borrado de cualquier almacenamiento sobre las observaciones para una medición concreta.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	Las observaciones serán del tipo textual y quedaran asociadas a cada alumno y a cada medición.

<b>OBJ-5</b>	Inserción y borrado de estímulos y factores en las actividades
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe permitir almacenar estímulos y/o factores para lograr realizar un seguimiento.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	

<b>OBJ-6</b>	Gestión de los alumnos
Versión	1.0
Autores	Moisés Martínez Mateu
Fuentes	ATE
Descripción	El sistema debe permitir asociar un alumno o grupos de alumnos a una determinada actividad.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	En Construcción
Estabilidad	Alta
Comentarios	Ninguno
<b>OBJ-7</b>	Autorización de usuarios

<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Fuentes</b>	ATE
<b>Descripción</b>	El sistema debe permitir la autorización para cada usuario registrado, ofreciéndole un rol.
<b>Subobjetivos</b>	Ninguno
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	En Construcción
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

### Catálogo de requisitos del sistema

En los siguientes apartados se listan los requisitos del sistema, tanto funcionales como no funcionales, y los diagramas de Casos de Uso.

#### Requisitos de Información

Los requisitos de información tienen como función reflejar el tipo de datos que tiene que manejar el sistema para cumplir cada uno de los objetivos. En nuestro caso, serán los siguientes:

<b>IRQ-1</b>	Información relacionado con los alumnos
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> <li>•</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada alumno
Datos específicos	<ul style="list-style-type: none"> <li>• ID del alumno</li> <li>• Nombre</li> <li>• Apellidos</li> <li>• Nombre de Padre/Madre</li> <li>• Fecha de nacimiento</li> <li>• Localidad</li> <li>• Provincia</li> <li>• Dirección</li> </ul>
Importancia	Alta
Estado	Implementado
Comentarios	Estos datos se encuentran recogidos en un servidor externo

<b>IRQ-2</b>	Información relacionado con los profesores
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> </ul>

	<ul style="list-style-type: none"> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> <li>•</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada profesor
Datos específicos	<ul style="list-style-type: none"> <li>• ID del profesor</li> <li>• Nombre</li> <li>• Apellidos</li> <li>• Especialidad</li> <li>• Fecha de nacimiento</li> <li>• Localidad</li> <li>• Provincia</li> <li>• Dirección</li> </ul>
Importancia	Alta
Estado	Implementado
Comentarios	Estos datos se encuentran recogidos en un servidor externo

IRQ-3	Información relacionado con las actividades
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> <li>•</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada actividad
Datos específicos	<ul style="list-style-type: none"> <li>• ID de la actividad</li> <li>• Nombre</li> <li>• Descripción</li> </ul>
Importancia	Alta
Estado	En construcción
Comentarios	Las actividades son aquellos ejercicios físicos o educativos que realizar los alumnos diariamente.

IRQ-4	Información relacionado con las observaciones
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> <li>• OBJ4 – Generación de informes.</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada observación
Datos específicos	<ul style="list-style-type: none"> <li>• ID de la observación</li> <li>• Título</li> </ul>



	<ul style="list-style-type: none"> <li>• Descripción</li> <li>• ID de la medición</li> </ul>
Importancia	Alta
Estado	En construcción
Comentarios	Las observaciones es una descripción extra que aporta detalles más técnicos sobre los datos recogidos en las mediciones.

IRQ-5	Información relacionado con las mediciones
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> <li>•</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada medición
Datos específicos	<ul style="list-style-type: none"> <li>• ID de la medición</li> <li>• Nombre</li> <li>• Fecha</li> <li>• Cuantía</li> <li>• Descripción</li> </ul>
Importancia	Alta
Estado	En construcción
Comentarios	Las mediciones son los datos que se obtienen por cada alumno en determinadas actividades. Por ejemplo, para una ATE controlar las necesidades fisiológicas de un alumno, el número de veces que un alumno va al baño en un día.

IRQ-6	Información relacionado con los estímulos
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición.</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada estímulo
Datos específicos	<ul style="list-style-type: none"> <li>• ID del estímulo</li> <li>• Título</li> <li>• Descripción</li> </ul>
Importancia	Alta

Estado	En construcción
Comentarios	
<b>IRQ-7</b>	<b>Información relacionado con los factores</b>
Objetivos asociados	<ul style="list-style-type: none"> <li>• OBJ1 - Inserción, modificación y borrado de mediciones.</li> <li>• OBJ2 - Insertar, modificar o borrar observaciones de cada medición</li> </ul>
Requisitos asociados	Ninguno
Descripción	El sistema debe almacenar la información correspondiente de cada factor
Datos específicos	<ul style="list-style-type: none"> <li>• ID del factor</li> <li>• Título</li> <li>• Descripción</li> </ul>
Importancia	Alta
Estado	En construcción
Comentarios	

### Requisitos Funcionales

Estos requisitos definen el comportamiento y los servicios que debe proveer el sistema:

<b>ACT-1</b>	<b>ATE</b>
Versión	1.0
Autores	Moisés Martínez Mateu
Descripción	Representa el conjunto de técnicos de atención especial
Comentarios	

<b>ACT-2</b>	<b>Servidor Web</b>
Versión	1.0
Autores	Moisés Martínez Mateu
Descripción	Representa el servidor web que realizara las correspondientes consultas a la base de datos

Comentarios

Diagramas de Casos de Uso

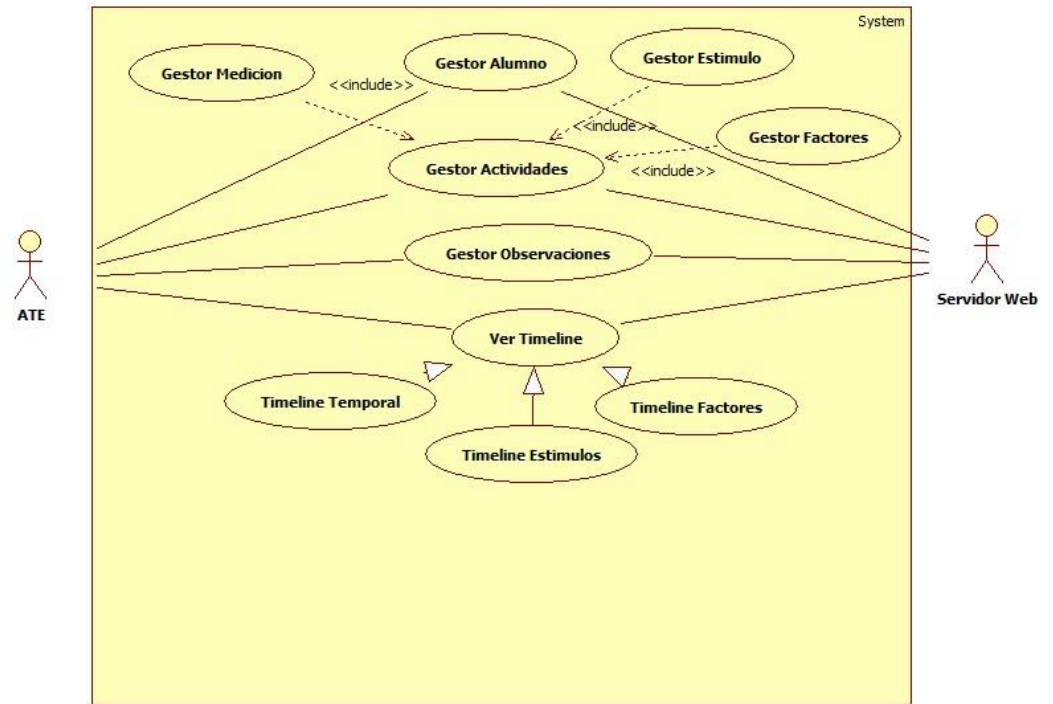


Ilustración 13- Caso de Uso del Sistema

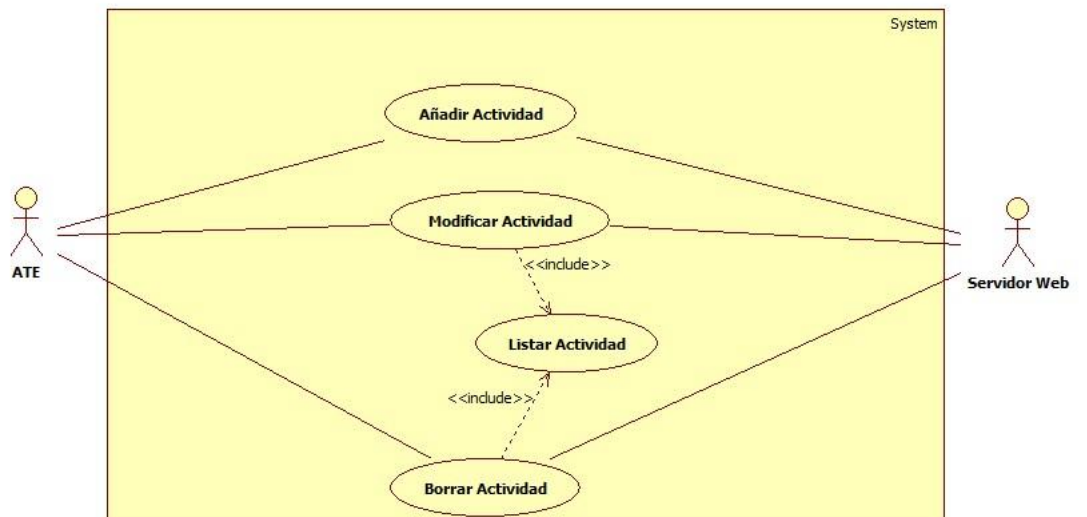


Ilustración 14- Caso de Uso Gestor Actividades

<b>UC-1</b>	<i>Alta Actividad</i>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 1 Gestión de Actividades OBJ- 7 Gestión de Alumnos	
<b>Requisitos asociados</b>	RI-1 Información relativa a las actividades	
<b>Descripción</b>	El sistema debe de permitir el alta de actividades por parte de los ATEs registrados previamente.	
<b>Precondición</b>	<i>Ninguna</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará el formulario de entrada
	<i>p2</i>	El ate introducirá los datos en el formulario y guardará el formulario
	<i>p3</i>	El sistema enviara el formulario al servidor web
	<i>p4</i>	El servidor web realiza la petición a la base de datos
	<i>p5</i>	<i>El sistema notifica el éxito de la operación</i>
<b>Postcondición</b>	<i>Actividad creada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p3 hay un error, se vuelve a p1
	<i>p2</i>	Si p4 hay un error, se ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

<b>UC-2</b>	<i>Modificar Actividad</i>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 1 Gestión de Actividades	
<b>Requisitos asociados</b>	RI-1 Información relativa a las actividades	
<b>Descripción</b>	El sistema debe de permitir la modificación de los datos propios de cada actividad	
<b>Precondición</b>	<i>Exista al menos una actividad</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado con las actividades disponibles
	<i>p2</i>	El ATE selecciona una actividad
	<i>p3</i>	El sistema recibirá los datos del servidor y autorrellenara los campos de texto

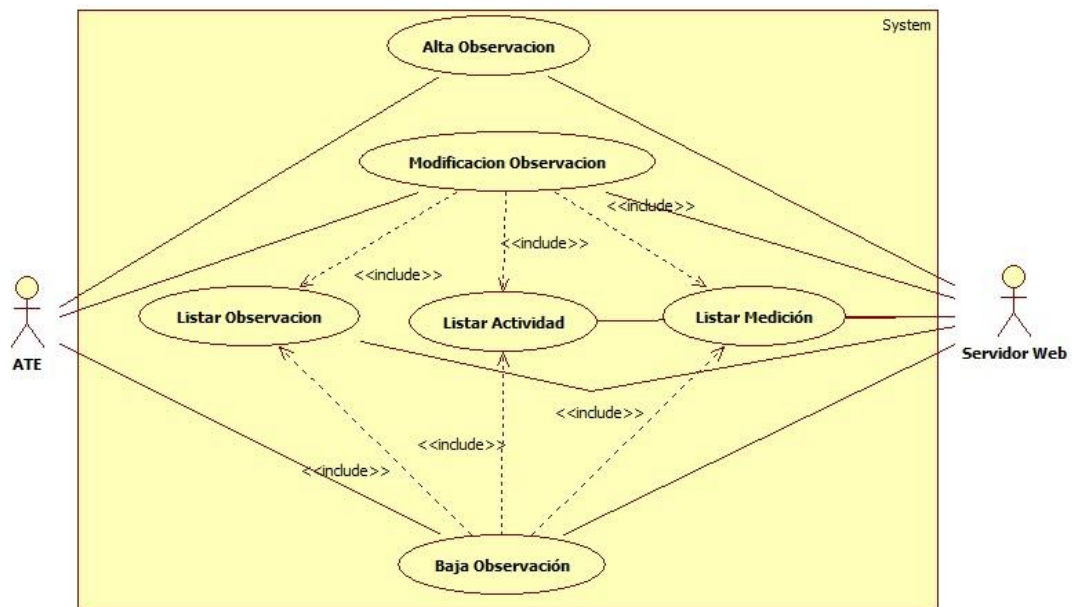
	<i>p4</i>	El ATE modificara los datos correspondientes y aceptará los cambios
<b>Postcondición</b>	<i>p5</i>	El sistema notifica el éxito de la operación
<b>Excepciones</b>	<i>Actividad modificada</i>	
	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

<b>UC-3</b>	<b><i>Borrar Actividad</i></b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 1 Gestión de Actividades	
<b>Requisitos asociados</b>	RI-1 Información relativa a las actividades	
<b>Descripción</b>	El sistema debe de permitir la modificación de los datos propios de cada actividad	
<b>Precondición</b>	<i>Exista al menos una actividad</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado con todas las actividades disponibles
	<i>p2</i>	El ATE selecciona una actividad y acepta la selección de borrado
	<i>p3</i>	El sistema recibirá los datos del servidor y procederá a la eliminación de ellos en el servidor
	<i>P4</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad modificada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	

**Estabilidad**  
**Comentarios**

*Alta*  
*Ninguno*

**CASO DE USO - GESTOR OBSERVACIONES**



<b>UC-4</b>	<i>Alta Observación</i>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 4 Gestor de Observaciones	
<b>Requisitos asociados</b>	RI – 4 Información relativa a las observaciones	
<b>Descripción</b>	El sistema debe de permitir la creación de nuevas observaciones para cada medición	
<b>Precondición</b>	<i>Ninguna</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado de mediciones
	<i>p2</i>	El ATE selecciona una medición y rellena los campos de texto sobre el título y descripción
	<i>p3</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos
	<i>P4</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad creada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1

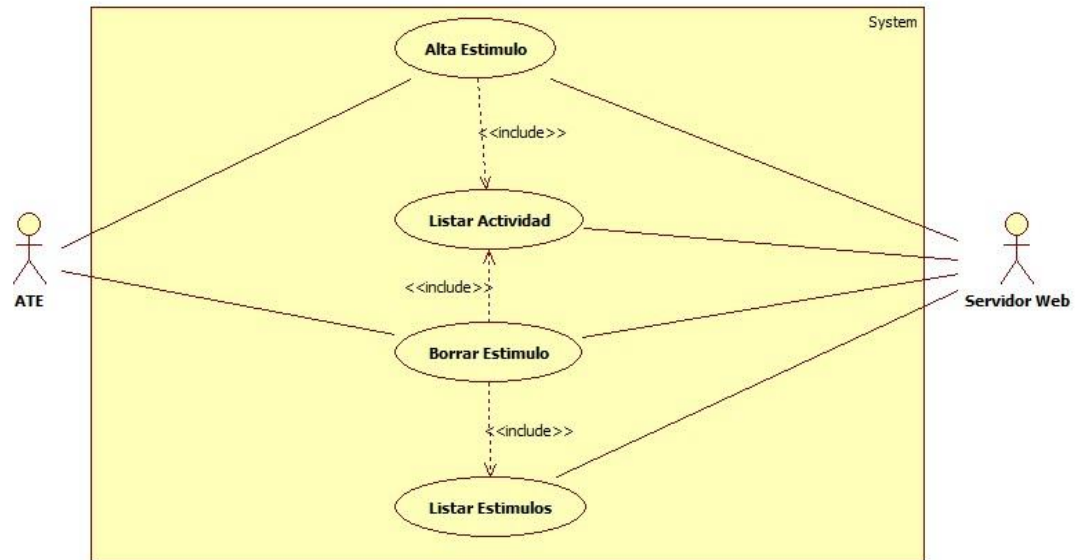
<b>Rendimiento</b>	...	...
	<b>Paso</b>	<b>Cota de tiempo</b>
	$q >$	$m <unidad de tiempo>$
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

<b>UC-5</b>	<b>Modificación Observación</b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 4 Gestor de Observaciones	
<b>Requisitos asociados</b>	RI – 4 Información relativa a las observaciones	
<b>Descripción</b>	El sistema debe permitir modificar los datos de las observaciones	
<b>Precondición</b>	<i>Exista al menos una observación</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado de actividades
	<i>p2</i>	El ATE selecciona una actividad
	<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad
	<i>P4</i>	El ATE selecciona una medición
	<i>P5</i>	El sistema rellenara los campos de texto con los datos originales
	<i>P6</i>	El ATE modifica los campos de texto
	<i>P7</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos
	<i>P8</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad creada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	$q >$	$m <unidad de tiempo>$
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	



<b>UC-6</b>	<b><i>Baja Observación</i></b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 4 Gestor de Observaciones	
<b>Requisitos asociados</b>	RI – 4 Información relativa a las observaciones	
<b>Descripción</b>	El sistema debe realizar la baja de las observaciones	
<b>Precondición</b>	<i>Exista al menos una observación</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado de observaciones disponibles
	<i>p2</i>	El ATE selecciona una observaciones y presiona el botón de borrar
	<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad
	<i>P4</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos
	<i>P5</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad borrada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

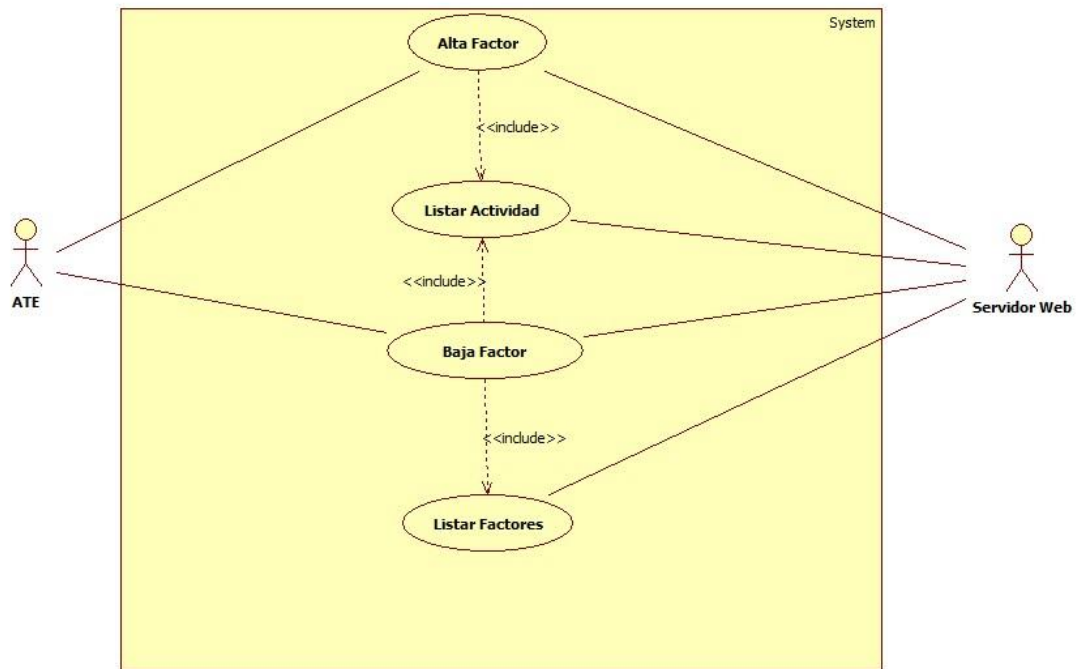
CASO DE USO – GESTOR DE ESTIMULOS



<b>UC-7</b>	<i>Alta Estimulo</i>												
<b>Versión</b>	1.0												
<b>Autores</b>	Moisés Martínez Mateu												
<b>Fuentes</b>	ATE												
<b>Objetivos asociados</b>	OBJ– 6 Asignación de Estímulos												
<b>Requisitos asociados</b>	RI – 7 Información relativa a los Estímulos												
<b>Descripción</b>	El sistema debe realizar la alta de los estímulos												
<b>Precondición</b>	<i>Exista al menos una actividad</i>												
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>El sistema presentará un listado de actividades disponibles</td> </tr> <tr> <td><i>p2</i></td> <td>El ATE selecciona una actividad</td> </tr> <tr> <td><i>p3</i></td> <td>El sistema presenta un listado de mediciones correspondientes con la actividad</td> </tr> <tr> <td><i>P4</i></td> <td>El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos</td> </tr> <tr> <td><i>P5</i></td> <td>El sistema notifica el éxito de la operación</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	El sistema presentará un listado de actividades disponibles	<i>p2</i>	El ATE selecciona una actividad	<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad	<i>P4</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos	<i>P5</i>	El sistema notifica el éxito de la operación
Paso	Acción												
<i>p1</i>	El sistema presentará un listado de actividades disponibles												
<i>p2</i>	El ATE selecciona una actividad												
<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad												
<i>P4</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos												
<i>P5</i>	El sistema notifica el éxito de la operación												
<b>Postcondición</b>	<i>Actividad borrada</i>												
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>Si p1 hay un error, se vuelve a p1</td> </tr> <tr> <td><i>p2</i></td> <td>Si p3 hay un error, notificara el error y ejecuta a p1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	Si p1 hay un error, se vuelve a p1	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1	...	...				
Paso	Acción												
<i>p1</i>	Si p1 hay un error, se vuelve a p1												
<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1												
...	...												
<b>Rendimiento</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Cota de tiempo</th> </tr> </thead> <tbody> <tr> <td><i>q &gt;</i></td> <td><i>m &lt;unidad de tiempo&gt;</i></td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Paso	Cota de tiempo	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>	...	...						
Paso	Cota de tiempo												
<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>												
...	...												
<b>Frecuencia</b>	<i>10/ 1 minuto</i>												
<b>Importancia</b>	<i>Alta</i>												
<b>Urgencia</b>	<i>Alta</i>												
<b>Estado</b>	<i>Urgente</i>												
<b>Estabilidad</b>	<i>Alta</i>												

<b>Comentarios</b>	<i>Ninguno</i>	
<b>UC-8</b>	<b><i>Borrar Estimulo</i></b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 6 Asignación de Estímulos	
<b>Requisitos asociados</b>	RI – 7 Información relativa a los Estímulos	
<b>Descripción</b>	El sistema debe realizar la baja de los estímulos creados	
<b>Precondición</b>	<i>Exista al menos una actividad</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado de estímulos disponibles
	<i>p2</i>	El ATE selecciona un estímulo
	<i>p3</i>	El sistema envía la petición al servidor, y esta procederá a la eliminación
	<i>P4</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad borrada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

CASO DE USO – GESTOR DE FACTORES

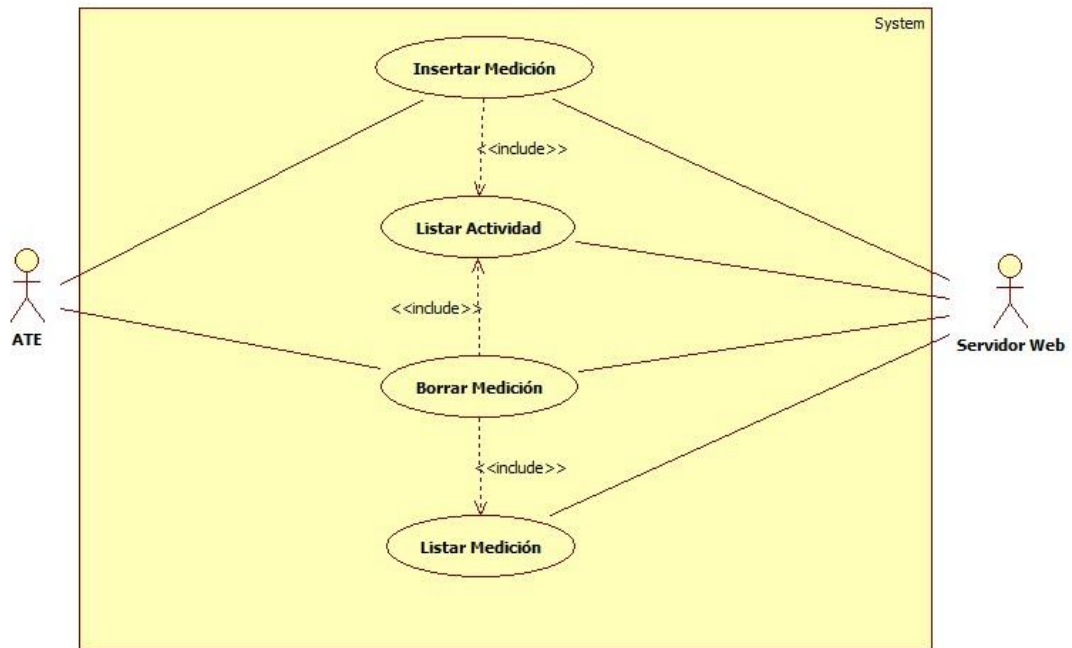


<b>UC-10</b>	<b>Alta Factor</b>												
<b>Versión</b>	1.0												
<b>Autores</b>	Moisés Martínez Mateu												
<b>Fuentes</b>	ATE												
<b>Objetivos asociados</b>	OBJ– 6 Asignación de Factores												
<b>Requisitos asociados</b>	RI – 8 Información relativa a los Factores												
<b>Descripción</b>	El sistema debe realizar la alta de los factores												
<b>Precondición</b>	<i>Exista al menos una actividad</i>												
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>El sistema presentará un listado de actividades disponibles</td> </tr> <tr> <td><i>p2</i></td> <td>El ATE selecciona una actividad</td> </tr> <tr> <td><i>p3</i></td> <td>El sistema presenta un listado de mediciones correspondientes con la actividad</td> </tr> <tr> <td><i>p4</i></td> <td>El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos</td> </tr> <tr> <td><i>p5</i></td> <td>El sistema notifica el éxito de la operación</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	El sistema presentará un listado de actividades disponibles	<i>p2</i>	El ATE selecciona una actividad	<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad	<i>p4</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos	<i>p5</i>	El sistema notifica el éxito de la operación
Paso	Acción												
<i>p1</i>	El sistema presentará un listado de actividades disponibles												
<i>p2</i>	El ATE selecciona una actividad												
<i>p3</i>	El sistema presenta un listado de mediciones correspondientes con la actividad												
<i>p4</i>	El sistema recibirá los datos del servidor y procederá a la alta de ellos en la base de datos												
<i>p5</i>	El sistema notifica el éxito de la operación												
<b>Postcondición</b>	<i>Actividad borrada</i>												
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>Si <i>p1</i> hay un error, se vuelve a <i>p1</i></td> </tr> <tr> <td><i>p2</i></td> <td>Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i></td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	Si <i>p1</i> hay un error, se vuelve a <i>p1</i>	<i>p2</i>	Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i>	...	...				
Paso	Acción												
<i>p1</i>	Si <i>p1</i> hay un error, se vuelve a <i>p1</i>												
<i>p2</i>	Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i>												
...	...												
<b>Rendimiento</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Cota de tiempo</th> </tr> </thead> <tbody> <tr> <td><i>q &gt;</i></td> <td><i>m &lt;unidad de tiempo&gt;</i></td> </tr> </tbody> </table>	Paso	Cota de tiempo	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>								
Paso	Cota de tiempo												
<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>												

<b>Frecuencia</b>	...	...
<b>Importancia</b>	<i>10/ 1 minuto</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Alta</i>	
<b>Estabilidad</b>	<i>Urgente</i>	
<b>Comentarios</b>	<i>Alta</i>	
	<i>Ninguno</i>	

<b>UC-11</b>	<b><i>Borrar Factor</i></b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ– 6 Asignación de Factores	
<b>Requisitos asociados</b>	RI – 8 Información relativa a los Factores	
<b>Descripción</b>	El sistema debe realizar la baja de los factores creados	
<b>Precondición</b>	<i>Exista al menos una actividad</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado de factores disponibles
	<i>p2</i>	El ATE selecciona un estímulo
	<i>p3</i>	El sistema envía la petición al servidor, y esta procederá a la eliminación
	<i>P4</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Actividad borrada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

CASO DE USO – GESTOR MEDICIÓN



<b>UC-12</b>	<b><i>Insertar Medición</i></b>														
<b>Versión</b>	1.0														
<b>Autores</b>	Moisés Martínez Mateu														
<b>Fuentes</b>	ATE														
<b>Objetivos asociados</b>	OBJ – 3 Inserción y borrado de mediciones														
<b>Requisitos asociados</b>	RI – 5 Información relativa de las mediciones														
<b>Descripción</b>	El sistema debe de permitir la inserción de mediciones para cada actividad														
<b>Precondición</b>	<i>Exista al menos una actividad</i>														
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>El sistema presentará un listado con las actividades disponibles</td> </tr> <tr> <td><i>p2</i></td> <td>El ATE selecciona una actividad</td> </tr> <tr> <td><i>p3</i></td> <td>El ATE rellena el campo de texto sobre el titulo e indicara los tiempos de inicio y final de la medición.</td> </tr> <tr> <td><i>p4</i></td> <td>El ATE debe rellenar el campo de descripción</td> </tr> <tr> <td><i>p5</i></td> <td>El sistema envía la petición al servidor, y esta procederá a la inserción</td> </tr> <tr> <td><i>P6</i></td> <td>El sistema notifica el éxito de la operación</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	El sistema presentará un listado con las actividades disponibles	<i>p2</i>	El ATE selecciona una actividad	<i>p3</i>	El ATE rellena el campo de texto sobre el titulo e indicara los tiempos de inicio y final de la medición.	<i>p4</i>	El ATE debe rellenar el campo de descripción	<i>p5</i>	El sistema envía la petición al servidor, y esta procederá a la inserción	<i>P6</i>	El sistema notifica el éxito de la operación
Paso	Acción														
<i>p1</i>	El sistema presentará un listado con las actividades disponibles														
<i>p2</i>	El ATE selecciona una actividad														
<i>p3</i>	El ATE rellena el campo de texto sobre el titulo e indicara los tiempos de inicio y final de la medición.														
<i>p4</i>	El ATE debe rellenar el campo de descripción														
<i>p5</i>	El sistema envía la petición al servidor, y esta procederá a la inserción														
<i>P6</i>	El sistema notifica el éxito de la operación														
<b>Postcondición</b>	<i>Medición creada</i>														
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td><i>p1</i></td> <td>Si <i>p1</i> hay un error, se vuelve a <i>p1</i></td> </tr> <tr> <td><i>p2</i></td> <td>Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i></td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Paso	Acción	<i>p1</i>	Si <i>p1</i> hay un error, se vuelve a <i>p1</i>	<i>p2</i>	Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i>	...	...						
Paso	Acción														
<i>p1</i>	Si <i>p1</i> hay un error, se vuelve a <i>p1</i>														
<i>p2</i>	Si <i>p3</i> hay un error, notificara el error y ejecuta a <i>p1</i>														
...	...														
<b>Rendimiento</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Cota de tiempo</th> </tr> </thead> <tbody> <tr> <td><i>q &gt;</i></td> <td><i>m &lt;unidad de tiempo&gt;</i></td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Paso	Cota de tiempo	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>	...	...								
Paso	Cota de tiempo														
<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>														
...	...														
<b>Frecuencia</b>	<i>10/ 1 minuto</i>														

<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>Urgente</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	<i>Ninguno</i>

<b>UC-12</b>	<b><i>Borrar Medición</i></b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Moisés Martínez Mateu	
<b>Fuentes</b>	ATE	
<b>Objetivos asociados</b>	OBJ – 3 Inserción y borrado de mediciones	
<b>Requisitos asociados</b>	RI – 5 Información relativa de las mediciones	
<b>Descripción</b>	El sistema debe de permitir la inserción de mediciones para cada actividad	
<b>Precondición</b>	<i>Exista al menos una actividad</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	El sistema presentará un listado con las actividades disponibles
	<i>p2</i>	El ATE selecciona una actividad
	<i>p3</i>	El ATE selecciona una medición y acepta los cambios
	<i>P4</i>	El sistema envía la petición al servidor, y esta procederá a la al borrado
	<i>P5</i>	El sistema notifica el éxito de la operación
<b>Postcondición</b>	<i>Medición borrada</i>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p1</i>	Si p1 hay un error, se vuelve a p1
	<i>p2</i>	Si p3 hay un error, notificara el error y ejecuta a p1
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	<i>q &gt;</i>	<i>m &lt;unidad de tiempo&gt;</i>
	...	...
<b>Frecuencia</b>	<i>10/ 1 minuto</i>	
<b>Importancia</b>	<i>Alta</i>	
<b>Urgencia</b>	<i>Alta</i>	
<b>Estado</b>	<i>Urgente</i>	
<b>Estabilidad</b>	<i>Alta</i>	
<b>Comentarios</b>	<i>Ninguno</i>	

**Requisitos No funcionales**

Los requisitos no funcionales guardan relación con todo aquello que no representa una funcionalidad concreta al sistema. Sino que fijan un atributo de calidad en el comportamiento de cada área del sistema.

<b>NFR-1</b>	<i>Alto desempeño de las conexiones (Rendimiento)</i>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Fuentes</b>	-----
<b>Objetivos asociados</b>	Todos los objetivos
<b>Requisitos asociados</b>	Todos los casos de uso ...
<b>Descripción</b>	El sistema deberá funcionar correctamente cuando los usuarios de un centro educativo deban conectarse y realizar las tareas cotidianas sin problemas graves en el servicio. Los servidores deben responder rápidamente y de forma correcta con cada petición.
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	En Construcción
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Implementar mecanismos de captura de excepciones

<b>NFR-2</b>	<i>Aspecto visual (Usabilidad)</i>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Fuentes</b>	-----
<b>Objetivos asociados</b>	Todos los objetivos
<b>Requisitos asociados</b>	Todos los casos de uso
<b>Descripción</b>	Al ser una aplicación para un dispositivo móvil hay que simplificar el uso de controles. Por ejemplo: El uso de calendarios para la gestión debe basarse en un timeline que contenga actividades, mediciones y observaciones de cada alumno en un curso determinado. De esta manera mejora la intuitividad del usuario aprovechando toda la pantalla, y al mismo tiempo, no es complicado de utilizar.
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	En Construcción
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	La interfaz es diferente para cada tipo de usuario

<b>NFR-3</b>	<i>Multidioma (Accesibilidad)</i>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Fuentes</b>	-----
<b>Objetivos asociados</b>	Todos los objetivos
<b>Requisitos asociados</b>	Todos los casos de uso
<b>Descripción</b>	La plataforma será multi-idioma, lo que permitirá la elección de diferentes lenguajes tales como el inglés, francés o alemán, a parte del español.
<b>Importancia</b>	Alta



<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construccion</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	<i>Usar internacionalización propia del Android SDK</i>

<b>NFR-4</b>	<i>Almacenamiento (Rendimiento)</i>
<b>Versión</b>	<i>1.0</i>
<b>Autores</b>	<i>Moisés Martínez Mateu</i>
<b>Fuentes</b>	<i>-----</i>
<b>Objetivos asociados</b>	<i>OBJ-3 Inserción, borrado de mediciones OBJ-2 Inserción, borrado y modificación de observaciones OBJ-1 Inserción, borrado y modificación de actividades OBJ-5 Inserción, borrado de estímulos OBJ-5 Inserción, borrado de factores</i>
<b>Requisitos asociados</b>	<i>IRQ-3 Datos relacionados con actividades IRQ-4 Datos relacionados con observaciones IRQ-5 Datos relacionados con mediciones IRQ-7 Datos relacionados con factores IRQ-6 Datos relacionados con estímulos</i>
<b>Descripción</b>	<i>Los requisitos de almacenamiento para la base de datos será del orden de GB para el sistema e irán aumentando de forma considerable con el número de usuarios (alumnos y profesores) y la información relativa a ellos.</i>
<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construcción</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	

<b>NFR-5</b>	<i>Copias de seguridad y mecanismos de respaldo (Seguridad)</i>
<b>Versión</b>	<i>1.0</i>
<b>Autores</b>	<i>Moisés Martínez Mateu</i>
<b>Fuentes</b>	<i>-----</i>
<b>Objetivos asociados</b>	<i>Todos los objetivos</i>
<b>Requisitos asociados</b>	<i>Todos los casos de uso ...</i>
<b>Descripción</b>	<i>Se diseñara un plan de seguridad que permita realizar backups incrementales los fines de semana o día festivo</i>
<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construcción</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	

<b>NFR-6</b>	<i>Open- Source</i>
<b>Versión</b>	<i>1.0</i>
<b>Autores</b>	<i>Moisés Martínez Mateu</i>
<b>Fuentes</b>	<i>-----</i>
<b>Objetivos asociados</b>	<i>Todos los objetivos</i>
<b>Requisitos asociados</b>	<i>Todos los casos de uso</i>
<b>Descripción</b>	<i>El sistema ofrece las ventajas de una tecnología open-source, así como un</i>

	coste menos frente a otras alternativas
<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construcción</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	-

<b>NFR-7</b>	<b><i>Roles y permisos (Seguridad)</i></b>
<b>Versión</b>	<i>1.0</i>
<b>Autores</b>	<i>Moisés Martínez Mateu</i>
<b>Fuentes</b>	-----
<b>Objetivos asociados</b>	<i>OBJ-7 Autorización de usuarios</i>
<b>Requisitos asociados _</b>	<i>IRQ-2 Datos relacionados con los profesores</i>
<b>Descripción</b>	El sistema provee un sistema de seguridad basado en permisos y roles, de los cuales cada usuario tendrá un rol diferente y cada uno tendrá un permiso (acceso) diferente tanto a los directorios del servidor como a las consultas en la base de datos.
<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construcción</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	

<b>NFR-8</b>	<b><i>Confiabledad de los datos (Seguridad)</i></b>
<b>Versión</b>	<i>1.0</i>
<b>Autores</b>	<i>Moisés Martínez Mateu</i>
<b>Fuentes</b>	-----
<b>Objetivos asociados</b>	Todos los objetivos
<b>Requisitos asociados _</b>	Todos los casos de uso
<b>Descripción</b>	Los diferentes elementos del sistema están asociados entre ellos con dispositivos externos como firewalls, switches con sistemas de prevención de ataques DoS y mecanismos para evitar ataques por SQL Injection o por PHP tunneling.
<b>Importancia</b>	<i>Alta</i>
<b>Urgencia</b>	<i>Alta</i>
<b>Estado</b>	<i>En Construcción</i>
<b>Estabilidad</b>	<i>Alta</i>
<b>Comentarios</b>	

### Diagrama de Clases

Un diagrama de clases tiene como objetivo facilitar una mejor comprensión de las relaciones entre las clases que componen un sistema. Además muestra las propiedades de los objetos y sus operaciones con un contexto y dominio especificado. Además también incluyen información sobre las relaciones entre objetos, incluso herencia.

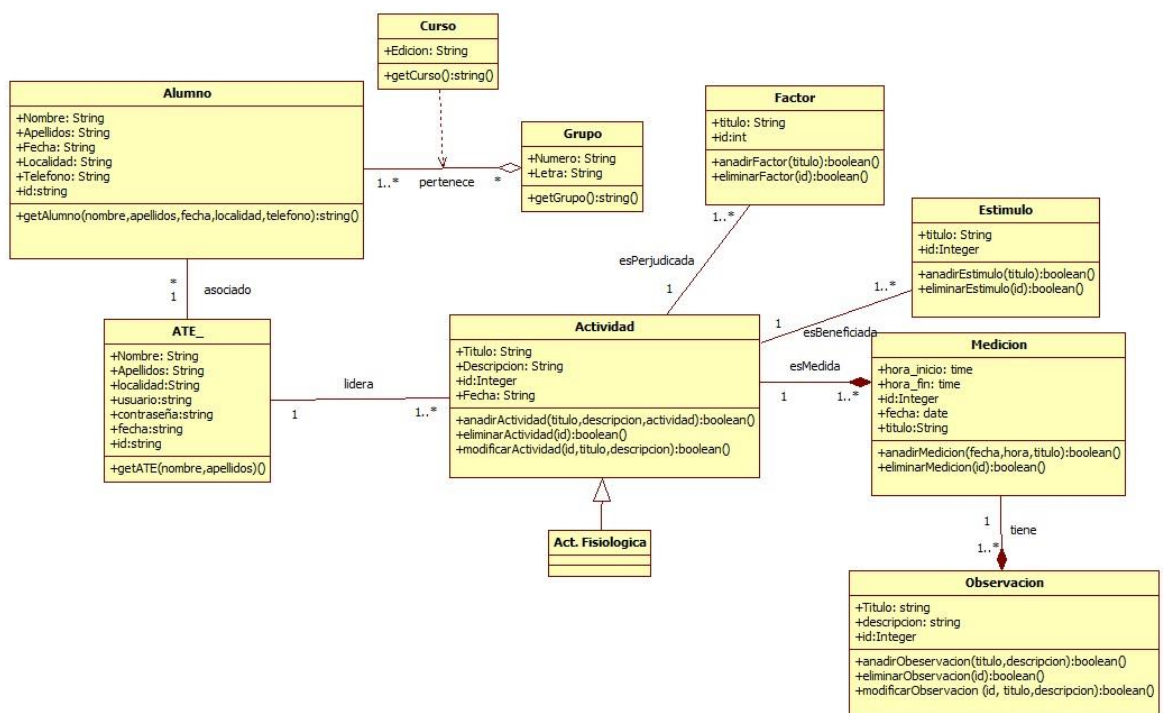


Ilustración 15- Diagrama de clases del sistema

#### Tipo de Objeto Alumno

<b>Tipo</b>	Alumno
<b>Versión</b>	1.0
<b>Autor</b>	Moisés
<b>Requisitos Asociados</b>	

<b>Descripción</b>	Este tipo representa cada uno de los alumnos pertenecientes en la escuela
<b>Comentarios</b>	

Atributos Objeto Alumno

<b>Atributo variable</b>	<b>Nombre</b>
<b>Descripción</b>	Representa el nombre de un alumno
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	""
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Apellidos</b>
<b>Descripción</b>	Representa los apellidos de un alumno
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	""
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Fecha</b>
<b>Descripción</b>	Representa la fecha de nacimiento de un alumno
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	""
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Localidad</b>
<b>Descripción</b>	Representa la localidad de residencia de un alumno
<b>Tipo OCL</b>	String

<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Dirección</b>
<b>Descripción</b>	Representa la dirección de residencia de un alumno
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Teléfono</b>
<b>Descripción</b>	Representa un teléfono de contacto
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

Tipo de Objeto Actividad

<b>Tipo</b>	<b>Actividad</b>
<b>Versión</b>	1.0
<b>Autor</b>	Moisés
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa cada una de las actividades que define un ATE sobre un alumno en particular.
<b>Comentarios</b>	

Atributos Objeto Actividad

<b>Atributo variable</b>	<b>Título</b>
<b>Descripción</b>	Representa el título identificativo de la actividad
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	Ejemplo: Ejercicio de Lectura, Comprensión Matemática

<b>Atributo variable</b>	<b>Descripción</b>
<b>Descripción</b>	Representa un breve texto para explicar el objetivo de la actividad.
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Id</b>
<b>Descripción</b>	Representa el parámetro identificativo de la actividad
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Fecha</b>
<b>Descripción</b>	Representa la fecha de creación de la actividad
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	Formato: dd/mm/aaaa

Tipo de Objeto ATE

<b>Tipo</b>	ATE
<b>Versión</b>	1.0
<b>Autor</b>	Moisés
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa cada uno de los ates en el sistema
<b>Comentarios</b>	

Atributos Objeto ATE

<b>Atributo variable</b>	Nombre
<b>Descripción</b>	Representa el nombre del ATE
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	""
<b>Comentarios</b>	

<b>Atributo variable</b>	Apellidos
<b>Descripción</b>	Representa los apellidos de los ATE
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	""
<b>Comentarios</b>	

Tipo de Objeto Medición

<b>Tipo</b>	<b>Medición</b>
<b>Versión</b>	1.0
<b>Autor</b>	Moisés
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa cada uno de las mediciones de cada actividad
<b>Comentarios</b>	

Atributos Medicion

<b>Atributo variable</b>	<b>Hora_inicio</b>
<b>Descripción</b>	Representa la hora de comienzo
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Hora_fin</b>
<b>Descripción</b>	Representa la hora final
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Fecha</b>
<b>Descripción</b>	Representa la fecha de creación de la medición
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	



<b>Atributo variable</b>	<b>Título</b>
<b>Descripción</b>	Representa un texto identificativo
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

Tipo de Objeto Observación

<b>Tipo</b>	<b>Observación</b>
<b>Versión</b>	1.0
<b>Autor</b>	Moisés
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa cada una de las observaciones asociadas a cada medicion
<b>Comentarios</b>	

Atributos Observación

<b>Atributo variable</b>	<b>Título</b>
<b>Descripción</b>	Representa el texto identificativo
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Descripción</b>
<b>Descripción</b>	Representa un breve texto explicativo
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>Id</b>
<b>Descripción</b>	Representa una clave de identificación
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	“”
<b>Comentarios</b>	

### Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. El diagrama de secuencia contiene detalles de implementación, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

### Login Usuario

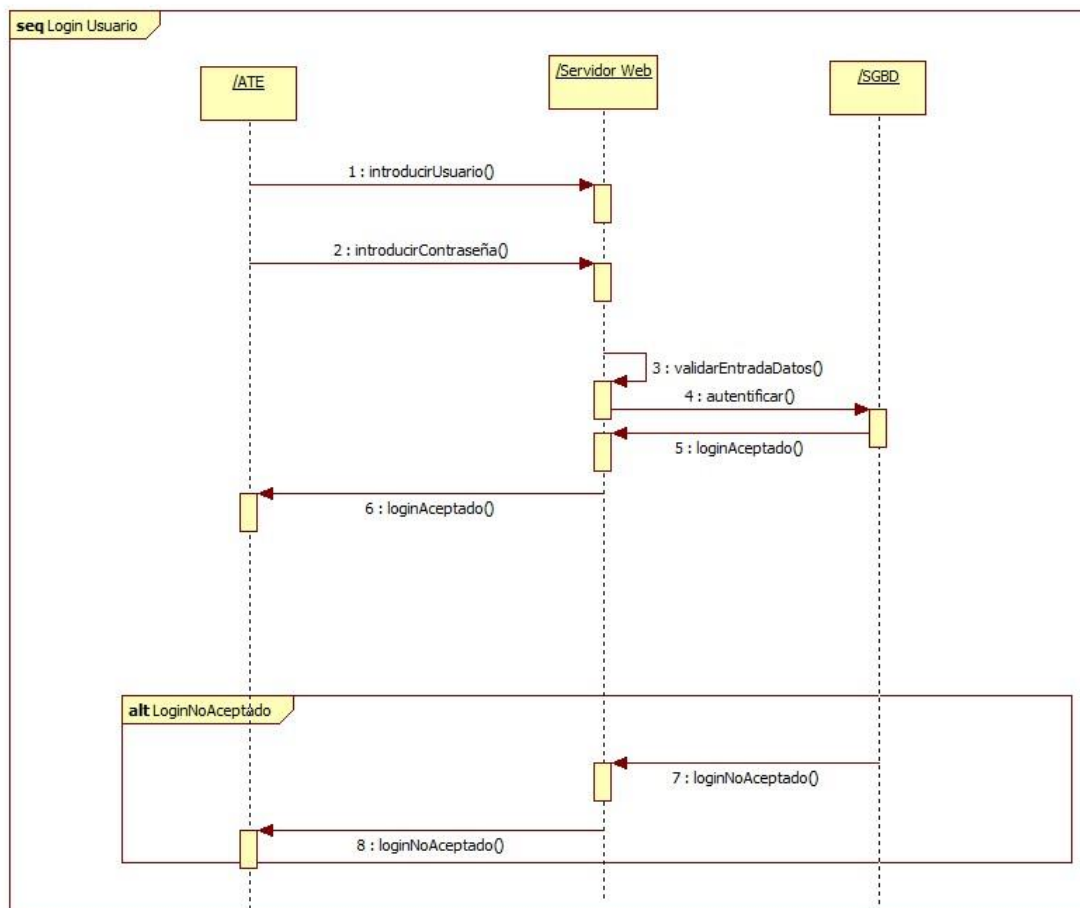


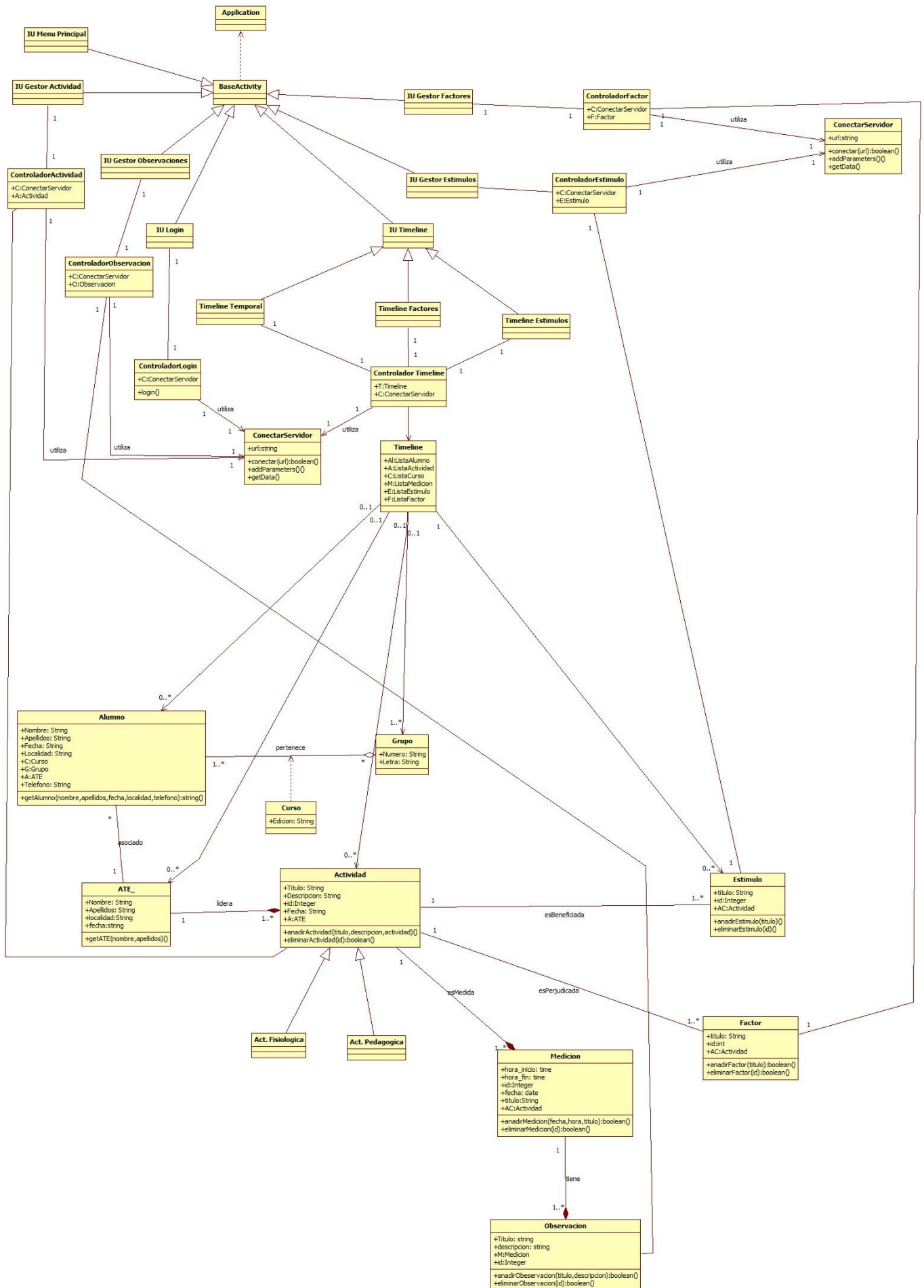
Ilustración 16- Diagrama de Secuencia Login

## **CAPÍTULO 4: DISEÑO DEL SISTEMA**

En este apartado, se tomarán todos los puntos relativos al diseño del proyecto software que abarcará, desde la definición hasta la especificación y diseño de la base de datos que se requiere para almacenar la información pertinente.

Así pues, en la etapa de diseño lo que se intenta hacer es acercar la idea conceptual que se ha obtenido de los requerimientos obtenidos en la fase de análisis, a un modelo más físico y más próximo a la futura implementación, que servirá de base para ésta. Siguiendo la estela de la especificación Android como arquitectura base.

## Diagrama de Clase



## Clase Actividad

Descripción de la clase Actividad

<b>Tipo</b>	<b>Actividad</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa una actividad
<b>Visibilidad</b>	Publica
<b>Supertipos</b>	-
<b>Subtipos</b>	-
<b>Comentarios</b>	Ninguno

Atributos de la clase Actividad

<b>Atributo variable</b>	<b>String::id</b>
<b>Descripcion</b>	Este atributo representa el id identificativo de cada actividad
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::titulo</b>
<b>Descripcion</b>	Este atributo representa el titulo de cada actividad
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::descripcion</b>
<b>Descripcion</b>	Este atributo representa la descripción de cada actividad



<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	
<b>Atributo variable</b>	<b>String::fecha</b>
<b>Descripcion</b>	Este atributo representa la fecha de creación de la actividad
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	
<b>Atributo variable</b>	<b>String::id_ate</b>
<b>Descripcion</b>	Este atributo representa el id del ate
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	
<b>Atributo variable</b>	<b>String::id_estimulo</b>
<b>Descripcion</b>	Este atributo representa el id del estimulo asociado
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	
<b>Atributo variable</b>	<b>String::id_factor</b>
<b>Descripcion</b>	Este atributo representa el id del factor asociado
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	
<b>Atributo variable</b>	<b>String::id_alumno</b>
<b>Descripción</b>	Este atributo representa el id del alumno
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

Métodos de la clase... Actividad

<b>Método de la clase miembro</b>	<b>HashMap&lt;Integer,Actividad&gt;::setMap()</b>
<b>Descripción</b>	Este método situara en un Hashmap cualquier referencia de la clase Actividad indexada por un id.
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno
<b>Método de la clase miembro</b>	<b>String::mostrarTituloActividad()</b>
<b>Descripción</b>	Este método devuelve el titulo de la actividad correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno
<b>Método de la clase miembro</b>	<b>String::mostrarIdActividad()</b>
<b>Descripción</b>	Este método devuelve el id de la actividad correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno
<b>Método de la clase miembro</b>	<b>HashMap &lt;Integer,String&gt;:: crearListaAdapter()</b>
<b>Descripción</b>	Este método crea un hashmap auxiliar donde parsea todos los datos de la original y los convierte a String
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno
<b>Método de la clase miembro</b>	<b>Void:: eliminarObjetoLista()</b>
<b>Descripción</b>	Este método elimina la referencia del objeto en el hashmap original
<b>Visibilidad</b>	Publica

<b>Comentarios</b>	Ninguno
--------------------	---------

*Asociación de clases... lidera (ATE\_, Actividad)*

Descripción de la Asociación.. lidera (ATE, Actividad)

<b>Asociación</b>	<b>Lidera entre ATE y Actividad</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta asociación representa la relación entre un ATE y una actividad asociada a él
<b>Comentarios</b>	Ninguno

Roles de la Asociación... lidera (ATE\_, Actividad)

<b>Rol</b>
<b>Descripción</b>
<b>Tipo OCL</b>
<b>Multiplicidad</b>
<b>Comentarios</b>

**Clase Alumno**

*Descripción de la clase Alumno*

<b>Tipo</b>	<b>Alumno</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa un alumno
<b>Visibilidad</b>	Publica
<b>Supertipos</b>	-
<b>Subtipos</b>	-
<b>Comentarios</b>	Ninguno

*Atributos de la clase Alumno*

Atributo variable	String::id
Descripcion	Este atributo representa el id identificativo de cada alumno
Tipo OCL	String
Visibilidad	Publica
Comentarios	

Atributo variable	String::nombre
Descripcion	Este atributo representa el nombre del alumno
Tipo OCL	String
Visibilidad	Publica
Comentarios	

Atributo variable	String::apellidos
Descripcion	Este atributo representa los apellidos del alumno
Tipo OCL	String
Visibilidad	Publica
Comentarios	

Atributo variable	String::fecha_nacimiento
Descripcion	Este atributo representa la fecha de nacimiento del alumno
Tipo OCL	String
Visibilidad	Publica
Comentarios	

Atributo variable	String::localidad
Descripcion	Este atributo representa la localidad del alumno
Tipo OCL	String
Visibilidad	Publica
Comentarios	

Atributo variable	String::direccion
Descripcion	Este atributo representa la dirección

	postal del alumno
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::telefono</b>
<b>Descripcion</b>	Este atributo representa teléfono de contacto del alumno
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

*Métodos de la clase... Alumno*

<b>Método de la clase miembro</b>	<b>HashMap&lt;Integer,Actividad&gt;::setMap()</b>
<b>Descripción</b>	Este método situara en un Hashmap cualquier referencia de la clase Alumno indexada por un id.
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarNombre()</b>
<b>Descripción</b>	Este método devuelve el titulo de la actividad correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarID()</b>
<b>Descripción</b>	Este método devuelve el id de la actividad correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarApellidos()</b>
-----------------------------------	-----------------------------------

<b>Descripción</b>	Este método devuelve los apellidos de un alumno
--------------------	---

<b>Visibilidad</b>	Publica
--------------------	---------

<b>Comentarios</b>	Ninguno
--------------------	---------

<b>Método de la clase miembro</b>	<b>HashMap &lt;Integer,String&gt;:: crearListaAdapter()</b>
-----------------------------------	---

<b>Descripción</b>	Este método crea un hashmap auxiliar donde parsea todos los datos de la original y los convierte a String
--------------------	---

<b>Visibilidad</b>	Pública
--------------------	---------

<b>Comentarios</b>	Ninguno
--------------------	---------

<b>Método de la clase miembro</b>	<b>Void:: eliminarObjetoLista()</b>
-----------------------------------	-------------------------------------

<b>Descripción</b>	Este método elimina la referencia del objeto en el hashmap original
--------------------	---

<b>Visibilidad</b>	Publica
--------------------	---------

<b>Comentarios</b>	Ninguno
--------------------	---------

*Asociación de clases... asociado (ATE\_ , Alumno)*

Descripción de la Asociación.. asociado (ATE, Alumno)

<b>Asociación</b>	<b>asociado entre ATE y Alumno</b>
-------------------	------------------------------------

<b>Versión</b>	1.0
----------------	-----

<b>Autores</b>	Moisés Martínez Mateu
----------------	-----------------------

<b>Requisitos Asociados</b>	
-----------------------------	--

<b>Descripción</b>	Esta asociación representa la relación entre un ATE y un alumno asociada a él
--------------------	---

<b>Comentarios</b>	Ninguno
--------------------	---------

Roles de la Asociación... asociado(ATE\_ , Alumno)

<b>Rol</b>	
------------	--

<b>Descripción</b>
<b>Tipo OCL</b>
<b>Multiplicidad</b>
<b>Comentarios</b>

### Clase Medición

#### *Descripción de la clase Medición*

<b>Tipo</b>	<b>Medicion</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa una medición
<b>Visibilidad</b>	Publica
<b>Supertipos</b>	-
<b>Subtipos</b>	-
<b>Comentarios</b>	Ninguno

#### *Atributos de la clase Medición*

<b>Atributo variable</b>	<b>String::id_actividad</b>
<b>Descripcion</b>	Este atributo representa el id de la actividad asociada a la medicion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::titulo</b>
<b>Descripcion</b>	Este atributo representa el nombre de la medicion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::descripcion</b>
<b>Descripcion</b>	Este atributo representa la descripcion de

	la medicion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::fecha</b>
<b>Descripcion</b>	Este atributo representa la fecha de creación de la medicion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::hora_inicio</b>
<b>Descripcion</b>	Este atributo representa la hora de inicio de la medición
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::hora_fin</b>
<b>Descripcion</b>	Este atributo representa la hora de fin de la medición
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

*Métodos de la clase... Medicion*

<b>Método de la clase miembro</b>	<b>HashMap&lt;Integer,Actividad&gt;::setMap ( )</b>
<b>Descripción</b>	Este método situara en un Hashmap cualquier referencia de la clase Medicion indexada por un id.
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno



<b>Método de la clase miembro</b>	<b>String::mostrarTitulo()</b>
<b>Descripción</b>	Este método devuelve el titulo de la medicion correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarIDObservacion()</b>
<b>Descripción</b>	Este método devuelve el id de la medicion
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>HashMap &lt;Integer,String&gt;:: crearListaAdapter()</b>
<b>Descripción</b>	Este método crea un hashmap auxiliar donde parsea todos los datos de la original y los convierte a String
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>Void:: eliminarObjetoLista()</b>
<b>Descripción</b>	Este método elimina la referencia del objeto en el hashmap original
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

*Asociación de clases... asociado (Actividad, Medición)*

Descripción de la Asociación.. tiene (Actividad, Medición)

<b>Asociación</b>	<b>Tiene entre Actividad y Medición</b>
-------------------	---

<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta asociación representa la relación entre un una actividad y una medicion
<b>Comentarios</b>	Ninguno

Roles de la Asociación... asociado(Actividad, Medicion)

<b>Rol</b>
<b>Descripción</b>
<b>Tipo OCL</b>
<b>Multiplicidad</b>
<b>Comentarios</b>

**Clase Observacion**

*Descripción de la clase Observacion*

<b>Tipo</b>	<b>Observacion</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Este tipo representa una observacion
<b>Visibilidad</b>	Publica
<b>Supertipos</b>	-
<b>Subtipos</b>	-
<b>Comentarios</b>	Ninguno

*Atributos de la clase Observacion*

<b>Atributo variable</b>	<b>String::id_medicion</b>
<b>Descripcion</b>	Este atributo representa el id de la medicion asociada a la observacion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::titulo</b>
<b>Descripcion</b>	Este atributo representa el nombre de la observacion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

<b>Atributo variable</b>	<b>String::descripcion</b>
<b>Descripcion</b>	Este atributo representa la descripcion de la observacion
<b>Tipo OCL</b>	String
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	

*Métodos de la clase... Observacion*

<b>Método de la clase miembro</b>	<b>HashMap&lt;Integer,Actividad&gt;::setMap()</b>
<b>Descripción</b>	Este método situara en un Hashmap cualquier referencia de la clase Observacion indexada por un id.
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarTitulo()</b>
<b>Descripción</b>	Este método devuelve el titulo de la observacion correspondiente
<b>Visibilidad</b>	Publica
<b>Comentarios</b>	Ninguno

<b>Método de la clase miembro</b>	<b>String::mostrarIDObservacion()</b>
<b>Descripción</b>	Este método devuelve el id de la observacion
<b>Visibilidad</b>	Publica

<b>Comentarios</b>	Ninguno
--------------------	---------

<b>Método de la clase miembro</b>	<b>HashMap &lt;Integer,String&gt;:: crearListaAdapter()</b>
<b>Descripción</b>	Este método crea un hashmap auxiliar donde parsea todos los datos de la original y los convierte a String
<b>Visibilidad</b>	Pública
<b>Comentarios</b>	Ninguno

*Asociación de clases... tiene (Medicion, Observacion)*

Descripción de la Asociación.. tiene (Medicion, Observacion)

<b>Asociación</b>	<b>Tiene entre Medicion y Observacion</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta asociación representa la relación entre un una medicion y una observacion
<b>Comentarios</b>	Ninguno

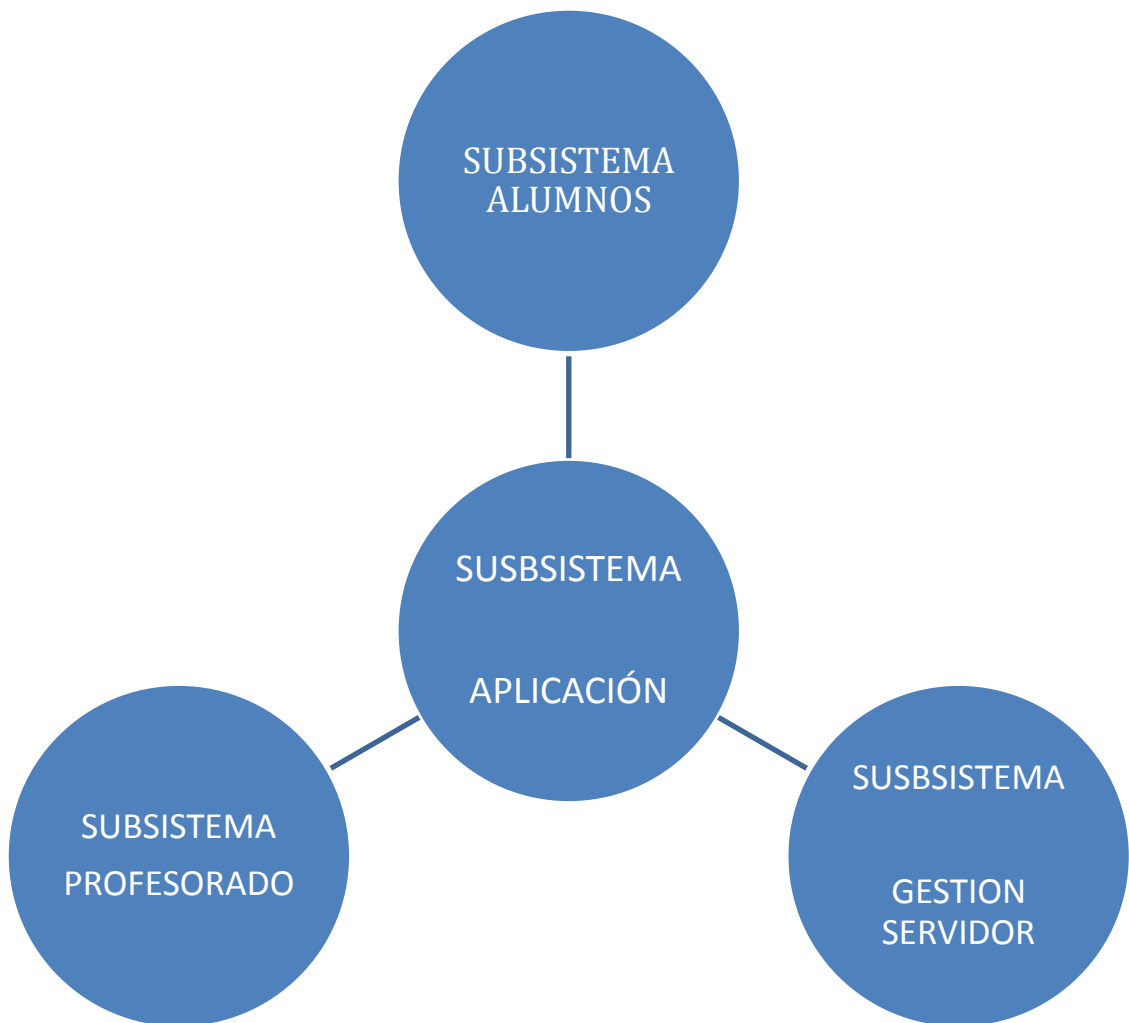
Roles de la Asociación... asociado(Actividad, Medicion)

<b>Rol</b>
<b>Descripción</b>
<b>Tipo OCL</b>
<b>Multiplicidad</b>
<b>Comentarios</b>

### Diseño General del Sistema

A continuación se muestra una visión global de la estructura de todo el sistema. Aquí se pueden observar los distintos subsistemas funcionales del sistema y las relaciones entre ellos.

Como se ha comentado anteriormente, la solución para la consecución del proyecto partía de una aplicación base. En esta visión global se incluyen tanto los subsistemas de la aplicación base, como los nuevos o modificados en el presente proyecto. Para diferenciarlos, se indican con fondo amarillo aquellos que se hayan tenido que modificar, desarrollar de cero o incluir:



Para poder identificar cada subsistema y la funcionalidad que cubre, se describe brevemente cada uno de ellos:

#### 1. Subsistema de Aplicación:

Es el subsistema principal. Inicialmente ya existía en la aplicación base, pero modificando éste es donde más se ha trabajado.

Éste es el que ofrece la mayor parte de las funcionalidades, así como el interfaz de usuario, y se apoya en el resto de los subsistemas incluidos en el diagrama, para poder proporcionarlas.

#### 2. Subsistema de Gestión de Servidor:

Es un subsistema que se encarga de realizar las peticiones HTTP al servidor web utilizando la API de Android sobre comunicaciones de redes. También se analizan los objetos JSON provenientes del servidor y se analizan/ parsean para que sean compatibles con Android.

Una vez descritos cada uno de los subsistemas que componen la aplicación completa, vamos a centrarnos en los dos subsistemas que tienen realmente impacto en el Trabajo Fin de Grado, que serían el Subsistema de Aplicación y el Subsistema de Base de Datos. Se presenta a continuación el diagrama de clases, donde tan solo se incluyen las clases afectadas:

### **Diseño de la Base de Datos**

#### Diseño de la base de datos

En este apartado se explicaran los detalles sobre el diseño que se basa la actual base de datos.

#### Modelo E/R

El modelo E/R representa la relación del conjunto de entidades que componen un sistema de información así como sus interrelaciones y sus propiedades. Esto ayudará a comprender mejor como se almacenarán los datos.

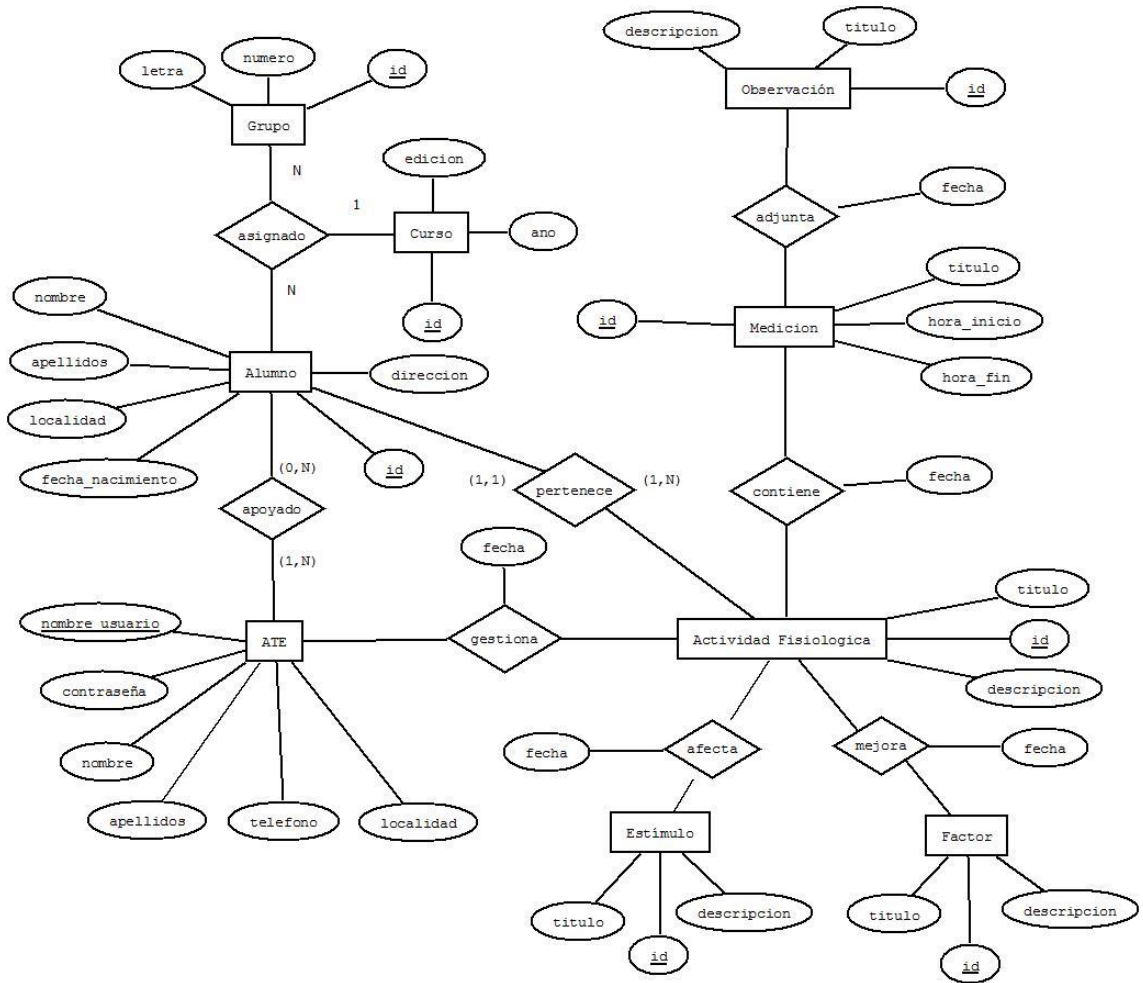


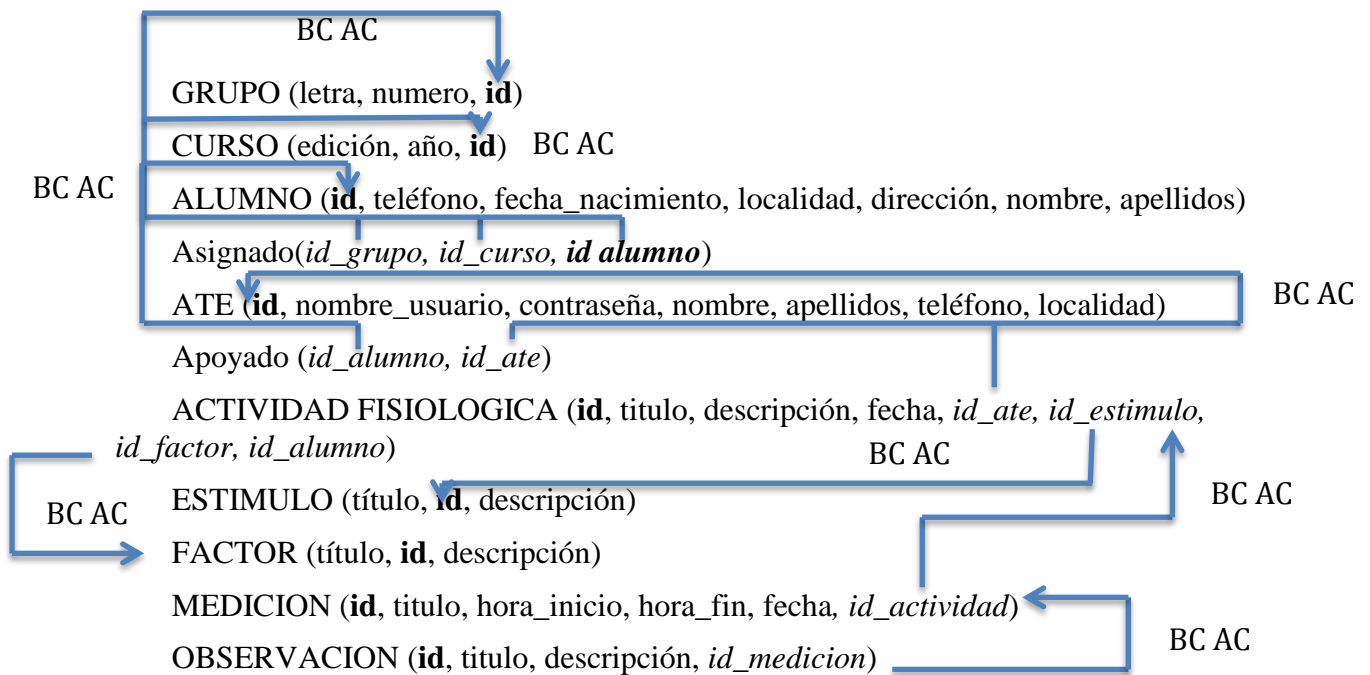
Ilustración 17- E/R Base de Datos

Modelo Relacional

El modelo relacional considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real.

Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional.





### *Descripción de la Base de Datos*

A continuación se detallan cada una de las entidades en el siguiente diccionario:

<b>Tipo</b>	<b>Base de Datos</b>		
<b>Versión</b>	1.0		
<b>Autores</b>	Moisés Martínez Mateu		
<b>Requisitos Asociados</b>			
<b>Descripción</b>	Esta es la base de datos utilizada por la aplicación AE-DROID		
<b>Tablas</b>	<b>Nombre</b>	<b>Tipo OCL</b>	<b>Mult</b>
	GRUPO	Tabla	
	CURSO	Tabla	
	ATE	Tabla	
	ACTIVIDAD FISIOLÓGICA	Tabla	
	MEDICION	Tabla	
	OBSERVACION	Tabla	
	ESTIMULO	Tabla	
	FACTOR	Tabla	
ALUMNO	Tabla		
<b>Comentarios</b>			

### *Tablas/Entidades de la Base de Datos*

- Descripción tabla **GRUPO**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de los grupos pertenecientes a cada curso escolar de primaria
<b>Comentarios</b>	

- Descripción atributos tabla **GRUPO**

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Valor</b>	<b>Clave</b>	<b>Clave</b>	<b>Restricciones</b>
--------------	-------------	--------------------	--------------	--------------	--------------	----------------------

			por Defecto	Primaria	Ajena	
<b>Letra</b>	VARCHAR		N/A	No	No	Not Null
<b>id</b>	INTEGER	Autonumérico	N/A	Si	No	Not Null
<b>Numero</b>	VARCHAR		N/A	No	No	Not Null

- Descripción tabla **CURSO**

Tipo	Tabla
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de los cursos escolares que se imparten en la escuela
<b>Comentarios</b>	

- Descripción atributos tabla **CURSO**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
<b>Edición</b>	VARCHAR			No	No	Not Null
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Año</b>	VARCHAR			No	No	Not Null

- Descripción tabla **ALUMNO**

Tipo	Tabla
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de los alumnos matriculados en la escuela
<b>Comentarios</b>	

- Descripción atributos tabla **ALUMNO**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
-------	------	-------------	-------------------	----------------	-------------	---------------

<b>Nombre</b>	VARCHAR		No	No	Not Null
<b>id</b>	INTEGER	Autonumerico	Si	No	Not Null
<b>Apellidos</b>	VARCHAR		No	No	Not Null
<b>Localidad</b>	VARCHAR		No	No	Not Null
<b>Fecha Nacimiento</b>	VARCHAR		No	No	Not Null
<b>Teléfono</b>	VARCHAR		No	No	Null
<b>Dirección</b>	VARCHAR		No	No	Not Null

- Descripción tabla **asignado**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena las relaciones entre los grupos, cursos y alumnos matriculados en el centro
<b>Comentarios</b>	

- Descripción atributos tabla **asignado**

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Valor por Defecto</b>	<b>Clave Primaria</b>	<b>Clave Ajena</b>	<b>Restricciones</b>
<b>Id_alumno</b>	VARCHAR			No	Si	CLAVE FORANEA
<b>id</b>	INTEGER	Autonumerico		Si	No	
<b>Id_curso</b>	VARCHAR			No	Si	CLAVE FORANEA
<b>Id_grupo</b>				No	Si	CLAVE FORANEA

- Descripción tabla **apoyado**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena la relación entre los alumnos y que Ate está asociado a él

**Comentarios**

- Descripción atributos tabla **apoyado**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
<b>Id_alumno</b>	INTEGER			No	Si	NOT NULL
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Id_ATE</b>	INTEGER			No	Si	NOT NULL

- Descripción tabla **ATE**

Tipo	Tabla
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de los ATE disponibles en el centro
<b>Comentarios</b>	

- Descripción atributos tabla **ATE**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
<b>Nombre_Usuario</b>	VARCHAR			No	No	NOT NULL
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Contraseña</b>	VARCHAR			No	No	NOT NULL
<b>Nombre</b>	VARCHAR			No	No	NOT NULL
<b>Apellidos</b>	VARCHAR			No	No	NOT NULL
<b>Localidad</b>	VARCHAR			No	No	NOT NULL
<b>Telefono</b>	VARCHAR			No	No	NULL

- Descripción tabla **ACTIVIDAD\_FISIOLOGICA**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de los actividades del tipo fisiológicos
<b>Comentarios</b>	

- Descripción atributos tabla **ACTIVIDAD\_FISIOLOGICA**

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Valor por Defecto</b>	<b>Clave Primaria</b>	<b>Clave Ajena</b>	<b>Restricciones</b>
<b>Nombre</b>	VARCHAR			No	No	NOT NULL
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Descripción</b>	VARCHAR			No	No	NULL
<b>Fecha</b>	DATETIME		CURRENT_TIMESTAMP	No	No	NOT NULL
<b>Id_estimulo</b>				No	Si	NOT NULL
<b>Id_factor</b>				No	Si	NOT NULL
<b>Id_ate</b>				No	Si	NOT NULL
<b>Id_alumno</b>				No	Si	NOT NULL

- Descripción tabla **ESTIMULO**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena todos los

estímulos de una actividad

**Comentarios**

- Descripción atributos tabla **ESTIMULO**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
Título	VARCHAR			No	No	NOT NULL
id	INTEGER	Autonumerico		Si	No	NOT NULL
Descripcion	VARCHAR			No	No	NULL

- Descripción tabla **FACTOR**

Tipo	Tabla
Versión	1.0
Autores	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
Descripción	Esta es la tabla que almacena todos los factores de una actividad
<b>Comentarios</b>	

- Descripción atributos tabla **FACTOR**

Campo	Tipo	Descripción	Valor por Defecto	Clave Primaria	Clave Ajena	Restricciones
Título	VARCHAR			No	No	NOT NULL
id	INTEGER	Autonumerico		Si	No	NOT NULL
Descripcion	VARCHAR			No	No	NULL

- Descripción tabla **MEDICION**

Tipo	Tabla
Versión	1.0
Autores	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
Descripción	Esta es la tabla que almacena cada uno de las mediciones realizadas en cada actividad

**Comentarios**

- Descripción atributos tabla **MEDICION**

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Valor por Defecto</b>	<b>Clave Primaria</b>	<b>Clave Ajena</b>	<b>Restricciones</b>
<b>Título</b>	VARCHAR			No	No	NOT NULL
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Id_actividad</b>	INTEGER			No	Si	NOT NULL
<b>Hora_inicio</b>	DATETIME			No	No	NOT NULL
<b>Hora_fin</b>	DATETIME			No	No	NOT NULL
<b>Fecha</b>	TIME			No	No	NOT NULL

- Descripción tabla **OBSERVACION**

<b>Tipo</b>	<b>Tabla</b>
<b>Versión</b>	1.0
<b>Autores</b>	Moisés Martínez Mateu
<b>Requisitos Asociados</b>	
<b>Descripción</b>	Esta es la tabla que almacena cada uno de las observaciones que se añaden a las mediciones presentes
<b>Comentarios</b>	

- Descripción atributos tabla **OBSERVACION**

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Valor por Defecto</b>	<b>Clave Primaria</b>	<b>Clave Ajena</b>	<b>Restricciones</b>
<b>Título</b>	VARCHAR			No	No	NOT NULL
<b>id</b>	INTEGER	Autonumerico		Si	No	NOT NULL
<b>Descripcion</b>	VARCHAR			No	No	NOT NULL
<b>Fecha</b>	DATETIME			No	No	NOT NULL

## Diseño de las interfaces de gráficas

Una ventaja de Android, respecto al diseño de los interfaces de Android, es que dispone de una herramienta de construcción de interfaces en el propio Android SDK, de esta manera desde el IDE se puede visualizar, editar cualquier prototipo de pantalla

En Android las pantallas se definen en archivos XML, donde se van incluyendo cada uno de los componentes del interfaz y sus propiedades.

Es importante no olvidar, que las aplicaciones Android van destinadas a un tipo de dispositivos que disponen de pantallas de reducido tamaño. Estas restricciones afectan a la cantidad de información mostrada, al tipo de componentes visuales que componen la ventana y al manejo de éstos (habitualmente táctil) que tiene una precisión mucho menor que los habituales punteros de un ordenador.

Teniendo en cuenta el número limitado número de horas del trabajo, se ha optado por unas interfaces sencillas y muy claras. A continuación se exponen los prototipos de las principales pantallas del sistema.



### Pantalla Inicio

En esta pantalla como se puede observar se disponen de dos campos de texto para los datos de acceso a la aplicación. Un botón para eventualizar la acción de conectarse al servidor y loguear al usuario o la salida de la aplicación.

El layout utilizado es del tipo LinearLayout, esto significa que cada componente queda anclado relativamente a los elementos más próximos: como otros botones, cajas de texto o el propio panel.



*Ilustración 18- Diseño pantalla de inicio*

### Pantalla Menú Principal

En esta pantalla tiene como objetivo disponer al usuario todas las opciones para el uso de las funcionalidades de la aplicación. Para ello se utiliza un LinearLayout con los botones con aquellas funciones importantes: Gestor Actividades, Gestor Observaciones, Timeline



*Ilustración 19- Diseño de la pantalla principal*

### Pantalla Gestor de Actividades

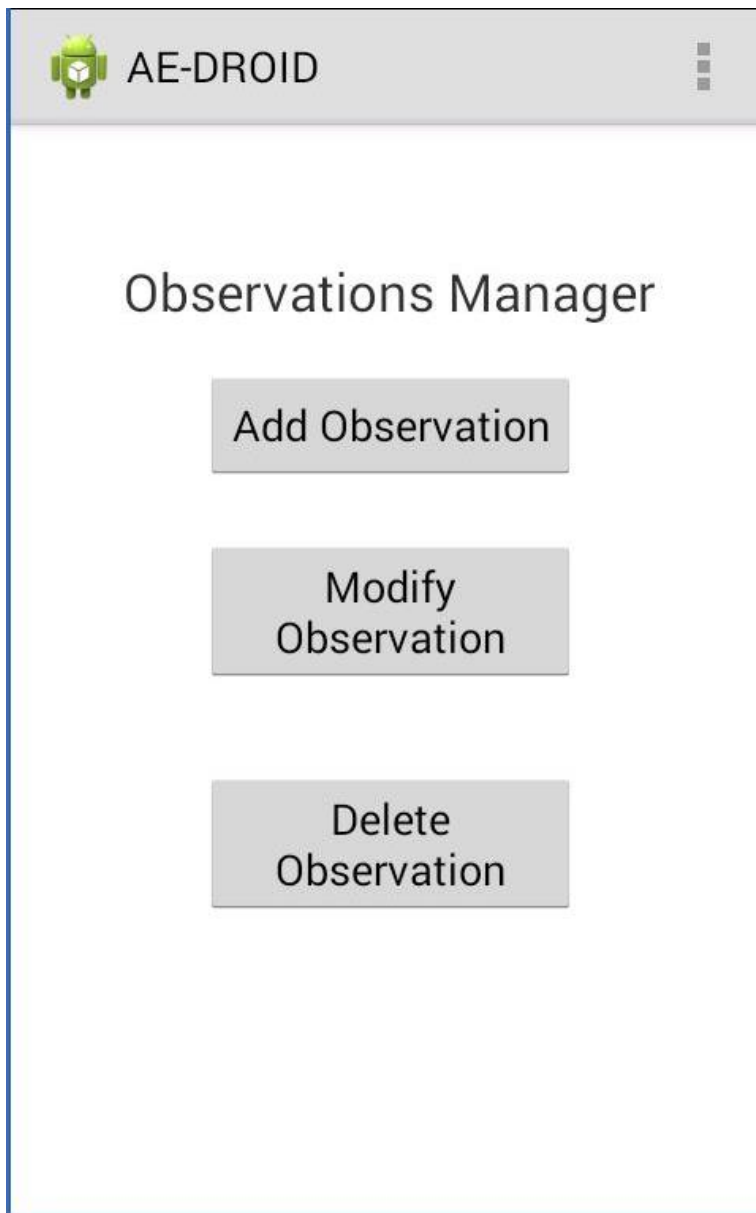
En esta pantalla se disponen de todas las operaciones relacionadas con la gestión de actividades. Siguiendo la misma línea de la pantalla anterior, tenemos los botones dispuestos para cada una de las funcionalidades.



*Ilustración 20- Diseño de la pantalla del gestor de actividades*

### Pantalla Gestor de Observaciones

En esta pantalla se disponen de todas las operaciones relacionadas con la gestión de observaciones. Siguiendo la misma línea de la pantalla anterior, tenemos los botones dispuestos para cada una de las funcionalidades.



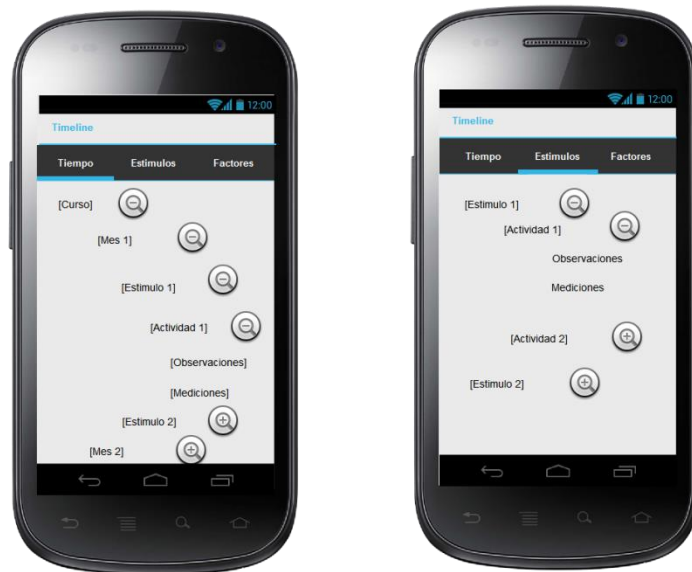
*Ilustración 21- Diseño de la pantalla del gestor de observaciones*

### Pantalla Timeline

En esta pantalla se sigue un estilo diferente usando un Tabhost o una vista de pestañas con la que podemos usar el mismo layout para cada tipo de ordenación diferente. El usuario solo tiene que seleccionar una pestaña y navegar a través del layout de forma sencilla y con una vista agradable sin artefactos.

Si se desea diseñar un layout atractivo es complicado dado que Android te ofrece muchas posibilidades, tantas que es muy posible dedicar varios meses solo para ello. La complejidad que conlleva ya la lógica de negocio imposibilita dedicar todo el tiempo al diseño.

Por eso se ha decantado por un diseño minimalista dentro de las posibilidades, intentando que el usuario no pierda pista de cada funcionalidad.



*Ilustración 22- Diseño del Timeline*

# **CAPÍTULO 5. PRUEBAS DEL SISTEMA**

## Pruebas y Validación

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde a un nivel distinto de involucramiento en las actividades de desarrollo.

Para las pruebas del sistema se han diseñado con el objetivo de supervisar la fiabilidad y estabilidad de la aplicación. Así como comprobar que el tratamiento del flujo alternativo es controlado y notificado correctamente a los usuarios.

Por tanto, el objetivo de esta fase es identificar el mayor número de errores posible, dentro de las limitaciones de tiempo y esfuerzo del proyecto. Para ello, sin duda, lo más conveniente es estructurar las pruebas y buscar los casos de pruebas más adecuados, para conseguir que esta fase sea lo más eficiente posible. La estructura de pruebas realizada es la siguiente:

- Pruebas Unitarias.

- **Caja Blanca:** encargadas de probar función a función, cada uno de los módulos de la aplicación.
- **Caja Negra:** encargadas de probar las funcionalidades, entradas y salidas de cada uno de los módulos de la aplicación.

- Pruebas de Integración.

Son las encargadas de probar la coherencia e integración entre módulos, es decir, si la llamadas entre ellos funcionan correctamente y la información intercambiada entre ellos es la esperada.

- Pruebas de Validación. Son las encargadas de probar el cumplimiento de los requerimientos del sistema.

- Pruebas de Sistema.

Son las encargadas de comprobar el comportamiento del sistema bajo situaciones donde es altamente ejercitado.

- Comprobación la fiabilidad de cada actividad

Cada actividad se prueba rellenando los campos con datos exagerados (por encima y por debajo de los límites establecidos).

- Comprobación la existencia de mensajes y mecanismos preventivos sobre errores
- Comprobación de los métodos de validación de los datos
- Análisis del rendimiento

Por tanto al tratarse de una actividad a posteriori, de detección y no de prevención, de problemas de software

### **Pruebas Unitarias**

Estas pruebas se han integrado con la fase de implementación, para asegurar un correcto funcionamiento de cada una de las funcionalidades, en cuanto finalizaba su desarrollo. De esta forma, se disponía de un alcance de las pruebas más acotado, que permitía comprobar tanto el software desarrollado, como el del proyecto base usado para alcanzar la funcionalidad.

### ***Caja Blanca***

Consiste en centrarse en la estructura interna del programa para elegir los casos de prueba. En este caso, la prueba ideal consiste en probar todos los posibles caminos de ejecución, a través de las instrucciones de código, que puedan trazarse.

Estas pruebas de Caja Blanca son necesarias para todo desarrollo software, pero no suficientes, ya que no permiten comprobar que el software hace lo que debe hacer, solo que lo que hace, lo hace bien.

### ***Caja Negra***

Consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos. Es ideal en esta prueba, verificar solo las entradas y las salidas sin entrar en detalle sobre su lógica interna.

Las pruebas de Caja Negra también se ejecutaron al finalizar el desarrollo de cada una de las funcionalidades, gracias a que la planificación se pensó para que el desarrollo fuera incremental y poder disponer así de los datos de entrada necesarios para poder probar cada módulo, pues el anterior ya estaba desarrollado y probado.

### **Pruebas de Integración**

Estas pruebas están totalmente ligadas a la forma prevista de integrar los distintos componentes del software hasta contar con el producto global que debe entregarse. Su objetivo fundamental es la prueba de las interfaces entre componentes o módulos

### **Prueba del Sistema**

La prueba del sistema es el proceso de prueba de un sistema integrado de hardware y software para comprobar si cumple los requisitos especificados, es decir:

- Se cumplen todos los requisitos funcionales, considerando el producto software final al completo, en un entorno del sistema.
- El funcionamiento y rendimiento en las interfaces de hardware, software, usuario y de operador.



- Adecuación de la documentación del usuario
- Ejecución y rendimiento en condiciones límite y de sobrecarga.

Centrándonos en la prueba realizada en este TFG, se utilizó un dispositivo móvil de última generación para comprobar que la aplicación funciona en los niveles correctos con el hardware real. También se midió correctamente el rendimiento general de la aplicación y su desempeño.

Los casos prueba usados eran pruebas de sobrecarga para comprobar la capacidad funcional de la aplicación en condiciones extremas y también utilizar técnicas de caja blanca

### Pruebas de Aceptación

El objetivo de estas pruebas consiste en comprobar si el producto está listo para ser implementado para el uso operativo en el entorno del usuario. Lo principal para evaluar el uso operativo del software suele ser la fiabilidad y la facilidad de uso del mismo. La prueba de aceptación es la prueba planificada y organizada formalmente para determinar si se cumplen los requisitos de aceptación marcados por el cliente.

Para el TFG se utilizó una segunda persona para comprobar la usabilidad y la fiabilidad en un entorno semi-operativo (bajo ciertas condicionantes). Ya que el objetivo es probar que el producto está listo para el uso operativo. Este usuario probó con total satisfacción todos los requisitos funcionales de la aplicación sin una planificación concreta. El usuario estuvo asesorado por el autor de este proyecto, Moisés Martínez, para probar también aquellos aspectos críticos como por ejemplo: las notificaciones sean perfectamente entendibles, que las transiciones entre pantallas sean correctas, que el manejo de la aplicación sea sencillo, etc.

### Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
26/08/14	1.0	Moisés		Pruebas Finales

### Información del Proyecto

Empresa / Organización	Universidad de Valladolid
Proyecto	AE-DROID: Aplicación para el seguimiento de necesidades especiales
Fecha de preparación	05/02/2014
Cliente	Universidad de Valladolid
Patrocinador principal	
Gerente / Líder de Proyecto	
Gerente / Líder de Pruebas de Software	

*Resumen Ejecutivo*

Es un resumen de todo el contenido del plan de Pruebas de Software, describe cuál es su propósito, establece si es un plan maestro o un plan detallado, identifica el alcance del plan de pruebas en relación con el plan de Proyecto de Software, restricciones (por ejemplo de recursos o presupuesto), alcance del esfuerzo de pruebas entre otros aspectos.

Para comenzar, detallamos el alcance del actual documento siempre trata de identificar aquellos aspectos funcionales del sistema, validarlos para garantizar que realizan correctamente su función con el objetivo así, de lograr un software robusto.

Se realizaran una serie de pruebas perfectamente identificadas y desarrolladas para garantizar un pase de calidad al cliente.

## Alcance de las Pruebas

### *Elementos de Pruebas*

Listado de todos los módulos, componentes o elementos que se van a probar. Si es de alto nivel, se listan las áreas funcionales (módulos o procesos que cubre el Testing), por otro lado, si es de un nivel detallado se listan los programas, unidades o módulos.

- **Clase LoginActivity.java:** Clase encargada de realizar el logueo con el servidor y de presentar al usuario la interfaz gráfica.
- **Clase AñadirMedicionActivity.java:** Clase encargada de añadir mediciones en una actividad y de presentar la correspondiente interfaz gráfica
- **Clase AñadirActividadActivity.java:** Clase encargada de añadir una actividad y de presentar la correspondiente interfaz gráfica.
- **Clase BorrarActividadActivity.java:** Clase encargada de borrar una actividad y de presentar la correspondiente interfaz gráfica.
- **Clase AnadirEstimuloActivity.java:** Clase encargada de añadir un estímulo a la actividad y de presentar la correspondiente interfaz gráfica
- **Clase AnadirObservacionActivity.java:** Clase encargada de crear una observación determinada a una medición y de presentar la correspondiente interfaz gráfica.
- **Clase ModificarObservacionActivity.java:** Clase encargada de modificar una observación y de presentar la correspondiente interfaz gráfica.
- **Clase BorrarObservacionActivity.java:** Clase encargada de borrar una actividad y de presentar la correspondiente interfaz gráfica.
- **Clase TimelineActivity.java:** Clase encargada de mostrar toda la información en árbol usando diferentes pestañas.

### *Nuevas Funcionalidades a Probar*

Las siguientes pruebas van a comprobar el correcto funcionamiento de las siguientes funcionalidades:

1. Comprobar el alta de las actividades
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca el alta en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
2. Comprobar el alta de las mediciones
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca el alta en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
3. Comprobar el alta de las observaciones
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca el alta en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
4. Comprobar el alta de los estímulos/factores
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca el alta en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
5. Comprobar el modificado de las actividades
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca la actualización en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
6. Comprobar el modificado de las observaciones
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca el actualización en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
7. Comprobar el borrado de las actividades
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca la baja en la correspondiente base de datos.

- c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
- 8. Comprobar el borrado de las observaciones
  - a. Verificar que los campos estén correctamente validados.
  - b. Verificar que se produzca la baja en la correspondiente base de datos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
- 9. Comprobar el timeline
  - a. Verificar que la correcta identificación de cada elemento.
  - b. Verificar que los datos mostrados sean los correctos.
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.
- 10. Comprobar el estado de red antes de realizar cualquier operación
  - a. Verificar que exista conexión de red y el servidor.
  - b. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.

### *Pruebas de Regresión*

Listado de las funcionalidades no directamente involucradas en el desarrollo, pero cuyos componentes están siendo afectados y por ende deben probarse para asegurar que continúan funcionando adecuadamente. Al igual que en el punto anterior, se describen desde el punto de vista del usuario.

- 1. Comprobar el timeline
  - a. Verificar que la correcta identificación de cada elemento
  - b. Verificar que los datos mostrados sean los correctos
  - c. Verificar que la aplicación debe notificar los correspondientes mensajes tanto en el flujo normal como en el alternativo.

### *Funcionalidades a No Probar*

Listado de las funcionalidades que NO se van a probar. Debe incluir información de las razones por las cuales no se van a probar y los riesgos que se están asumiendo.

- La funcionalidad acerca del borrado de mediciones no se va a probar, se prescinde porque utiliza prácticamente el mismo código que el borrado de observaciones
- La funcionalidad acerca del borrado de estímulos/factores no se va a probar

### *Enfoque de Pruebas (Estrategia)*

La Estrategia de Pruebas puede definirse como un documento aparte, o puede ser incluido dentro del Plan de Pruebas según su extensión. Aquí pueden definirse los tipos de pruebas a realizar (funcionales, de desempeño, de interfaces, no funcionales, etc.), requerimientos especiales de las pruebas, configuraciones a probar, subconjuntos de datos a considerar, nivel de pruebas de regresión, entre otros aspectos.

Las pruebas a realizar serán del tipo funcionales y de desempeño:

- **Funcionales:** Estudiarán el comportamiento de las tareas más comunes que tendrán que realizar los usuarios.
- **De Desempeño:** Estudiarán el rendimiento de la aplicación y de cada una de las tareas más críticas del sistema por parte de los usuarios.

Las configuraciones tanto del servidor como del dispositivo móvil deben ser las mismas que en la fase de producción.

También se debe considerar la tipología de los datos que coincidan con los límites establecidos en la fase de diseño de la base de datos. Cualquier campo ha de ser probado y por supuesto, que el paso de parámetros entre la capa cliente y la capa servidora sea realmente en UTF-8, así se evitará, caracteres anómalos que puedan provocar una pérdida de consistencia y coherencia en todo el sistema.

### **Criterios de Aceptación o Rechazo**

Son los criterios que serán considerados para dar por completado el Plan de Pruebas de Software, por ejemplo: Completar 100% de pruebas unitarias, cierto porcentaje de casos exitosos, cobertura de todos los componentes y líneas de código, porcentaje de defectos corregidos, entre otros.

Para seguir con el plan de calidad del proyecto se debe asegurar que se aprueba al 100% todas las unidades de prueba desarrolladas. Los casos de prueba ya sean unitarios, de integración, de sistema, de aceptación

## CAPÍTULO 6. CONCLUSIONES PERSONALES

Para garantizar la buena eficacia de todo el trabajo de desarrollo de este proyecto, hay que evaluar al alumno en el contexto de aprendizaje, valorando aquello que es capaz de hacer el alumno y todos los factores que facilitan o dificultan el aprendizaje.

La identificación de necesidades educativas es un complejo y meticuloso proceso en el que intervienen multitud de elementos, los cuales se tienen que elegir bajo la opción más correcta dependiendo de la necesidad que se tenga, así se utilizará un tipo de evaluación u otra, ya sea ordinaria, multidimensional, asistida, etc. Debemos recordar que es importante centrarse en el proceso, es decir, buscar las necesidades educativas en el proceso interactivo de enseñanza/ aprendizaje. Así, consideramos de vital importancia la evaluación psicopedagógica donde se analiza todo el proceso de enseñanza aprendizaje, detecta las NEE y toma las decisiones curriculares respecto a la escolarización de estos alumnos.

El desarrollo de este proyecto ha implicado un gran esfuerzo no solo en comprender este nuevo entorno, sino también aprender cómo se aplican las nuevas tecnologías como JSON o Android a los modernos dispositivos móviles. Los componentes gráficos que ofrece Android y su correcta utilización no es tarea sencilla pues requiere de muchas horas utilizando layout, adaptadores o listas especiales para cada uno.

Por ello afrontar el reto de realizar un timeline que sirva para realizar un seguimiento estándar y tenerlo acabado en el tiempo límite del proyecto, sin apenas conocimientos de Android ha sido más que satisfactorio.



## CAPÍTULO 7. FUTURAS MEJORAS

Según se han explicitado las bases de los objetivos recientes de este proyecto, el diseño actual del sistema permite las siguientes mejoras:

1. Añadir la posibilidad de que el sistema exporte informes e infografías de los datos que se almacenan, para apoyar el seguimiento de los profesionales. La exportación podría realizarse para ser compatible con los formatos Word y PDF.

2. Otros profesionales del sector como profesores o directores puedan realizar u observar también el seguimiento que se está realizando con el objetivo de mejorar la eficacia

3. Añadir la posibilidad de que los tutores puedan concertar reuniones con otros tutores, así como los padres de los alumnos quedando en constancia para el informe de evaluación de cada uno de ellos

4. Implementar un chat interno en la propia aplicación móvil para que cualquier profesional pueda comunicarse entre ellos. Incluyendo la posibilidad de cada mensaje pueda ser supervisado por el director del centro.

## CAPÍTULO 8. DEFINICIONES

### Modelo Peticiones – Respuesta

Para emplear la tecnología JSON para un webserver es necesario el empleo de métodos que establezcan conexión con un servidor web y este a su vez se lo comunique a un sistema gestor de bases de datos. Es muy recomendable el empleo de un middleware que sirva de nexo de control entre la aplicación cliente y la base de datos por prevención de riesgos por cortes en la red y pueda sufrir la base de datos problemas de inconsistencia.

Aparte de eso utilizar el middleware sirve también para realizar una doble validación de los datos antes de enviarlos a la base de datos. Claro está, en detrimento del rendimiento global por la adicción de este tercero al conjunto.

Como he dicho, basándonos en un servidor web, el empleo de un modelo petición-respuesta es mucho más eficaz para aplicaciones distribuidas.

### JSON

JSON (JavaScript Object Notation) es un formato específico de datos ligeros. Cuyo objetivo es ser fácil de entender por parte de los humanos, al mismo tiempo fácil de interpretar y parsear en una computadora

JSON está dividida en dos partes:

- Una colección de nombre/valor.
- Una lista ordenada de valores.

Aunque originalmente derivó del lenguaje de script Javascript, JSON es un lenguaje independiente del formato, y es compatible con multitud de lenguajes de programación

Los tipos de básicos de JSON:

- Un *objeto* es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termine con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).

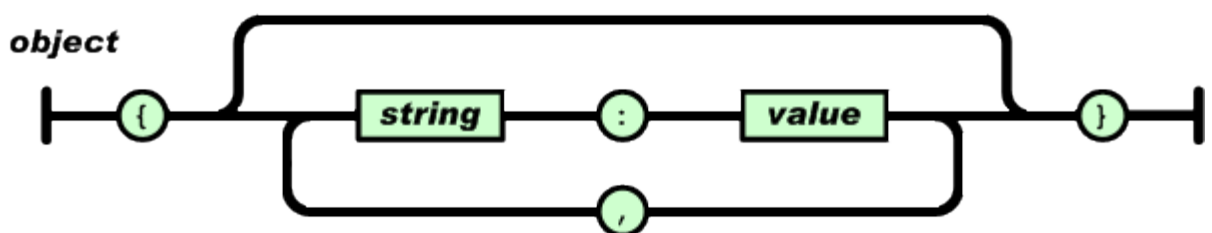


Ilustración 23- Representación de un objeto JSON

- Un *arreglo* es una colección de valores. Un arreglo comienza con [ (corchete izquierdo) y termina con ] (corchete derecho). Los valores se separan por , (coma).

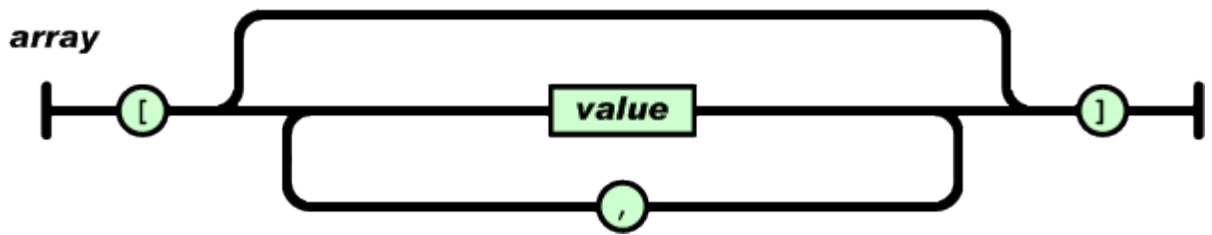


Ilustración 24- Representación de un array en JSON

- Un *valor* puede ser una *cadena de caracteres* con comillas dobles, o un *número*, o true o false o null, o un *objeto* o un *arreglo*. Estas estructuras pueden anidarse.

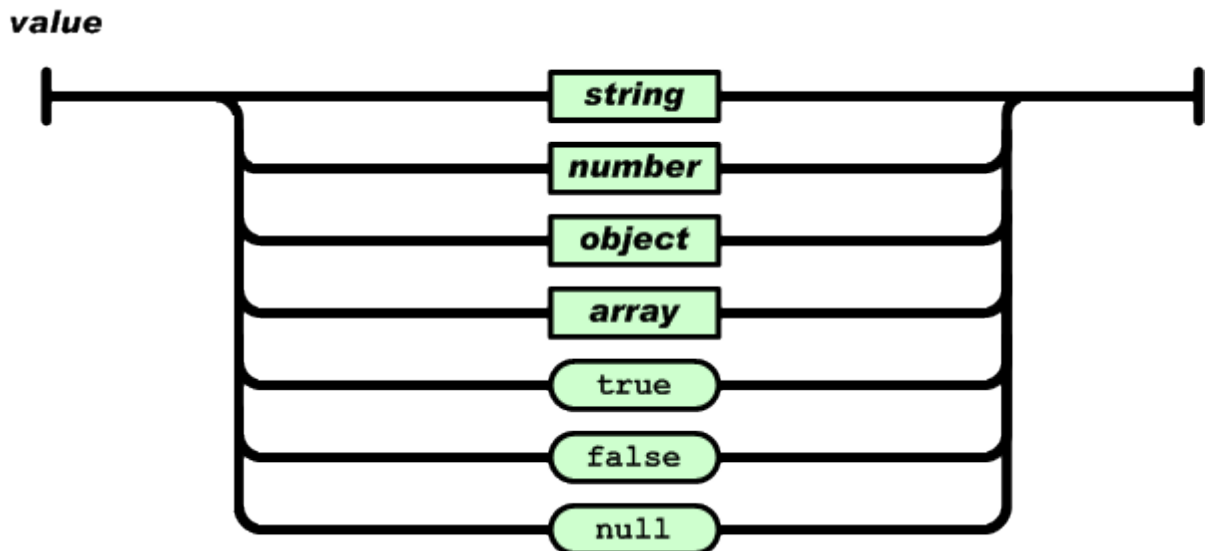


Ilustración 25- Posibles valores de un objeto JSON

- Una *cadena de caracteres* es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter. Una *cadena de caracteres* es parecida a una cadena de caracteres C o Java.

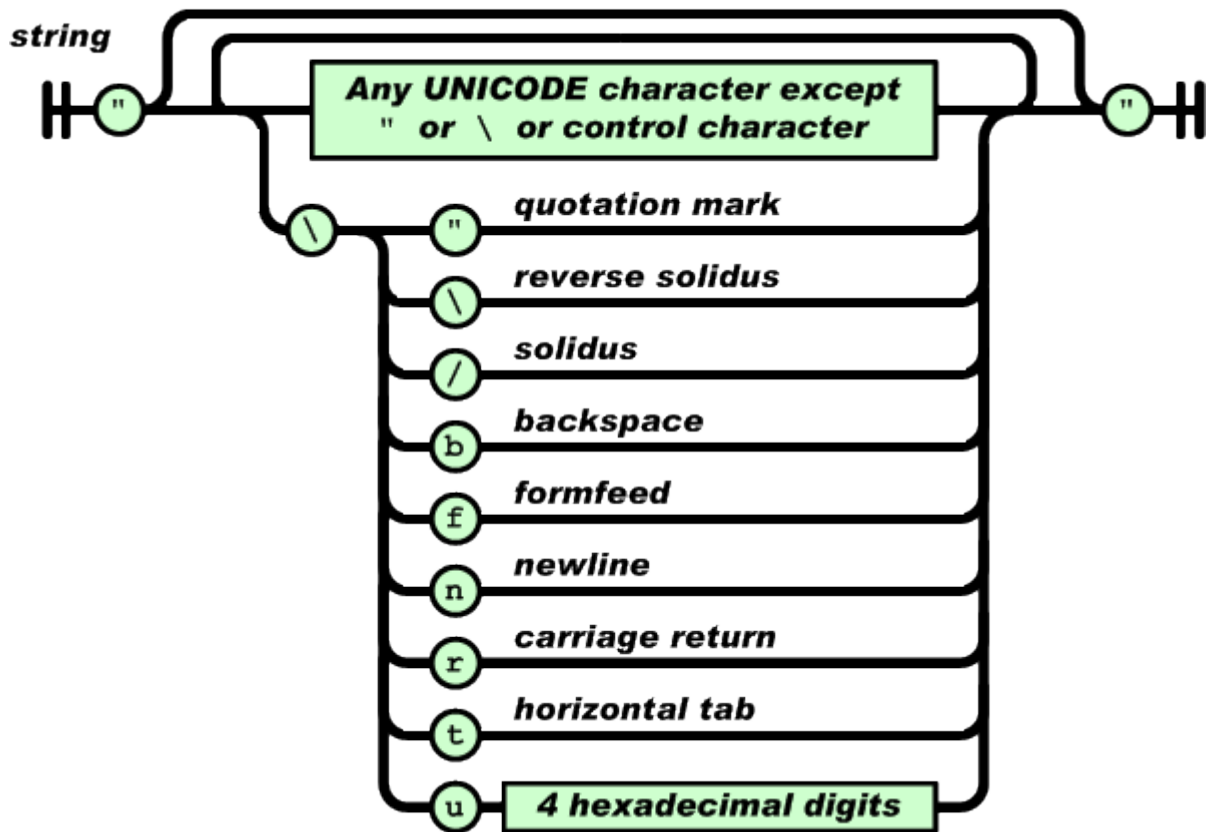


Ilustración 26- Cadena de caracteres de un objeto JSON

- Un *número* es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.

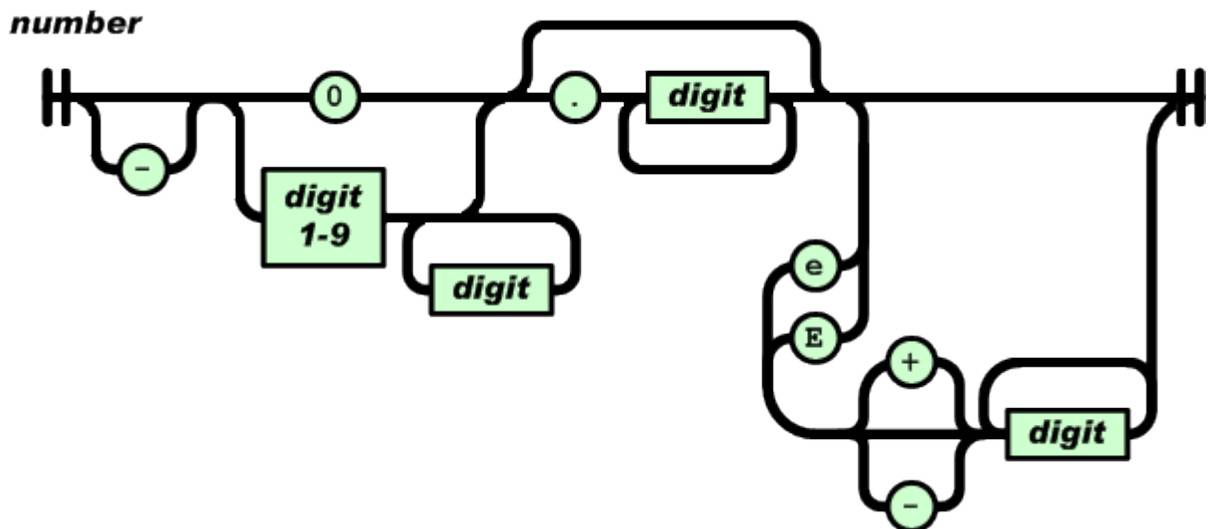


Ilustración 27- Rango de números que puede tomar un objeto JSON

Los espacios en blanco pueden insertarse entre cualquier par de símbolos.

### Middleware

El término es más comúnmente utilizado para el software que permite la comunicación y la gestión de datos en aplicaciones distribuidas. Este paradigma servirá como guía cliente-servidor.

Funciona como una capa de abstracción de software distribuida, que se sitúa entre dos capas arquitectónicas. El middleware abstrae de la complejidad y heterogeneidad de las redes de comunicaciones, así de detalles complejos, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas.

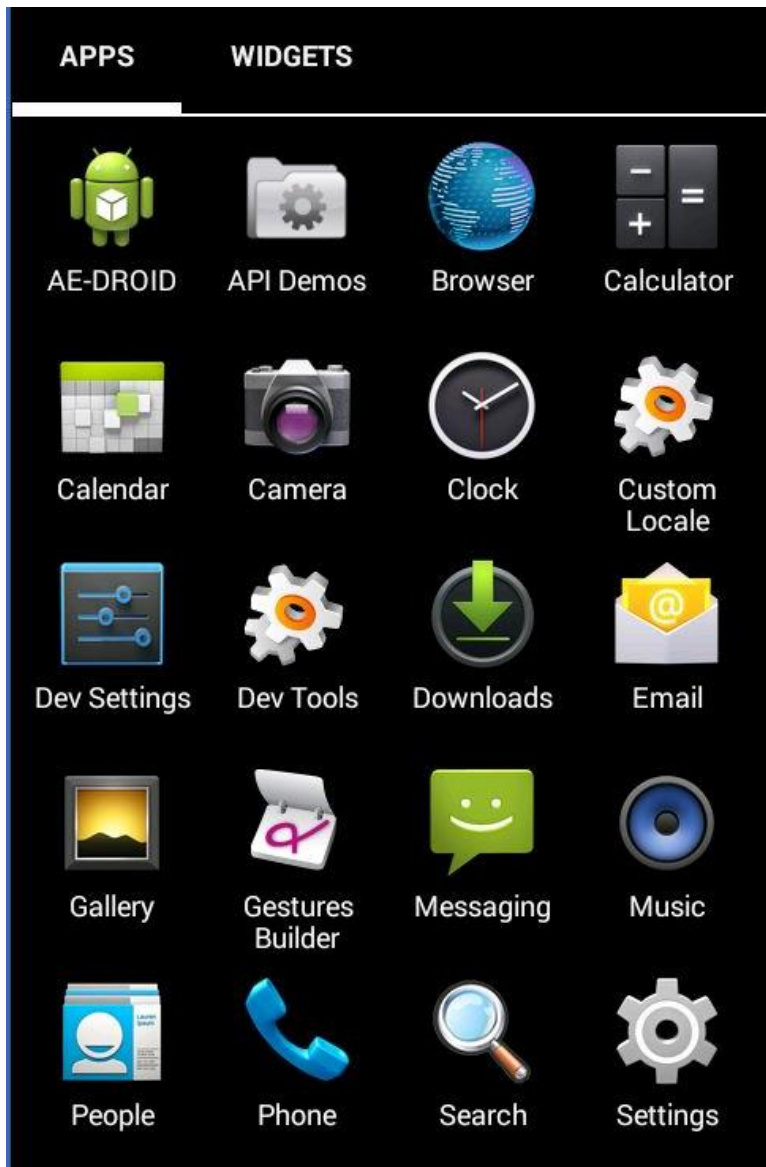
Los middleware que se encuentran fuertemente unidos a la red se llaman servidores de aplicaciones, ya que mejoraran el rendimiento, disponibilidad, escalabilidad, seguridad, recuperación de información, y soportan la administración colaborativa y su uso. Los middleware pueden contactar directamente a la aplicación ganando mejor comunicación entre el servidor y el cliente. Otros servicios importantes dados por este tipo de middleware son servicios de directorios, emails, cadenas de suministros de gran tamaño, accesos remotos a información, descarga de archivos, accesos a programas y acceso a aplicaciones remotas.

## **SECCION 2. MANUAL DE USUARIO**

## Manual de Instalación

### Cliente

Para instalar la aplicación cliente en cualquier dispositivo móvil Android, hay que instalar el paquete .apk desde la memoria interna del dispositivo con ayuda de un gestor de archivos particular y posteriormente aceptar los términos. El sistema debe crear un acceso directo, como muestra la siguiente imagen:



*Ilustración 28- Acceso directo a la aplicación*

Una vez terminado, faltaría modificar la IP de conexión del programa para que conecte con el servidor correcto.

Al ser un proyecto educativo, se recomienda utilizar el IDE Eclipse junto con el Android SDK y ejecutar dicho proyecto desde allí.

## Servidor

En el apartado de servidor, hay que instalar los archivos auxiliares en el directorio htdocs a la raíz del servidor. Dependiendo de qué plataforma Apache estemos utilizando este directorio puede ser htdocs o www.

Una vez terminada la subida de ficheros al servidor, basta con iniciar los servicios de inicio del servidor en el sistema operativo.

Si por algún motivo existe algún problema en la conexión, revise los datos de conexión a la base de datos en estos archivos.

## Base de datos

En primer lugar iniciar los servicios de arranque de mysql (mysqld.exe) para inicializar todo el sistema que permita el acceso/consulta a la base de datos.

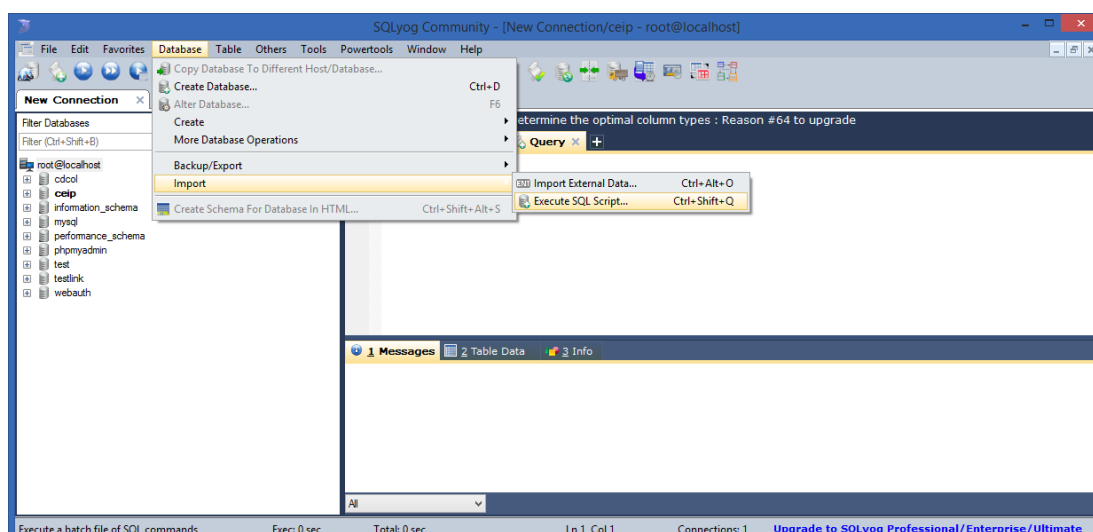
El usuario normal debe ser: **root y sin contraseña**. Para cambiar estos datos hay que modificar el fuente de los archivos del servidor.

En segundo lugar importar el fichero **ceip.sql** al sistema gestor de base de datos utilizando la consola de mysql o utilizando algún gestor externo como phpmyadmin. En este caso, se utilizó una alternativa gratuita para plataformas Windows, **SQLyog Community Edition**. SQLyog es un potente gestor de grandes bases de datos relacionales con el apoyo de más de 30.000 clientes, en su edición gratuita, escrita en C++, nos permitirá hacer las tareas fundamentales de consulta, inserción y diseño de la base de datos.

Cabe destacar que en la versión de pago es ideal cuando necesitar herramientas integradas sobre optimización y análisis de rendimiento

Para importar una base de datos:

1. En la barra menú superior, seleccionar el menú Database, posteriormente Import y por último Execute SQL Script.

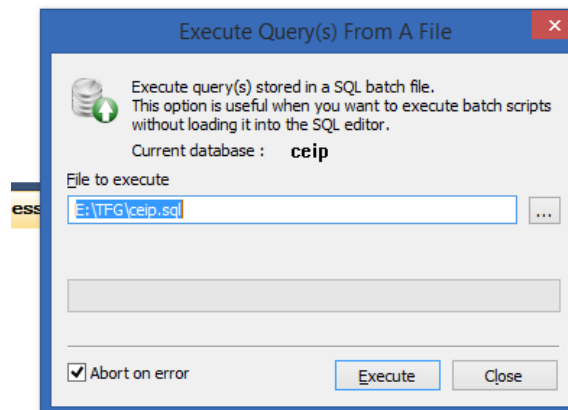


*Ilustración 29- Interfaz gráfica SQLyog*



2. Aparecerá una ventana de dialogo solicitando el directorio origen del fichero SQL. Seleccionamos el directorio donde se encuentra el fichero y finalmente le damos al botón Execute.

Si durante el proceso existiese algún error, el programa abortaría la importación y mostraría que errores se han producido.



*Ilustración 30- Importación de fichero SQL*

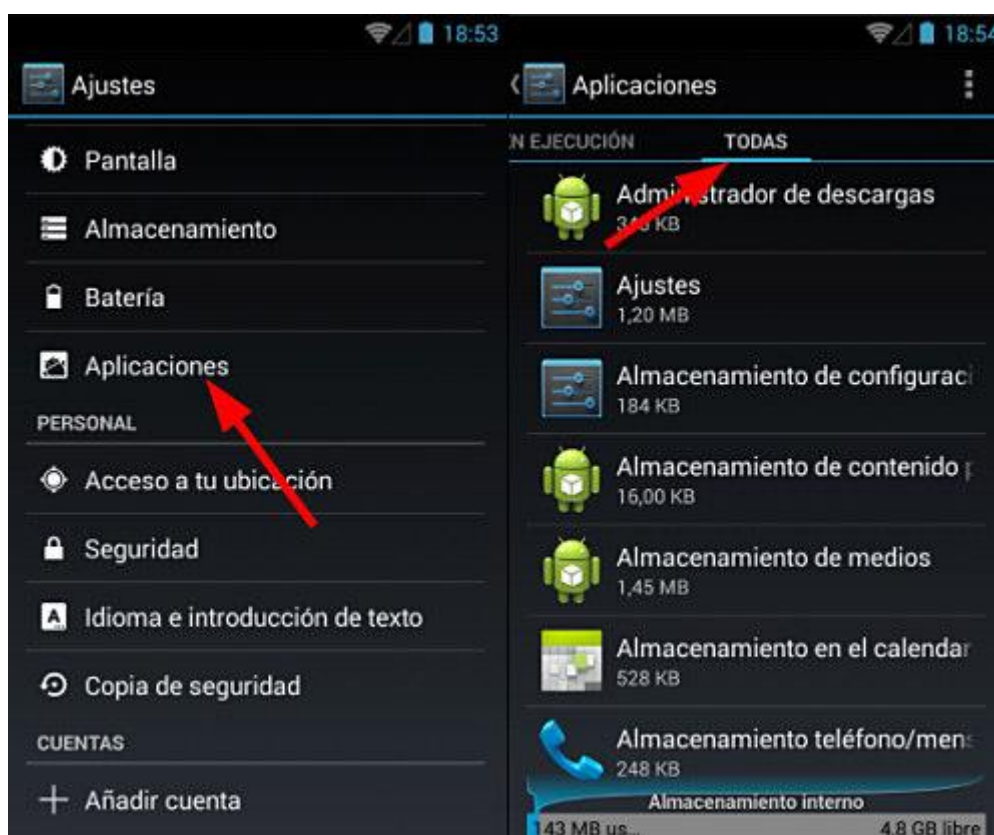
**En los ficheros fuente, es necesario configurar algunos datos para el correcto funcionamiento de todos los componentes. Tales como los datos de acceso a la base de datos, dispuestos en cada fichero clase.**

## Manual de Desinstalación

Para desinstalar la aplicación desde el sistema Android se procede como en cualquier aplicación normal o bien se puede utilizar el software propio para desinstalar aplicaciones o bien utilizar un software de terceros como Titanium Backup.

Para desinstalar aplicaciones con el software propio de Android hay que realizar los siguientes pasos (la versión mostrada es Android 4.0):

1. Para comenzar, lo primero que tenemos que hacer es desplazarnos hasta el ítem “**Aplicaciones**” ubicado en el apartado “**Ajustes**”. Luego de ello, pulsamos sobre la pestaña “**Todas las aplicaciones**”.
2. En este punto, se desplegará una lista con todos los programas que tenemos instalados en nuestro teléfono Android, incluyendo las apps de sistema, **por lo cual debemos ser precavidos al momento de seleccionar algo para su desinstalación.**



*Ilustración 31- Listado de aplicaciones instaladas y menú de ajustes*

1. En esta lista buscaremos la aplicación en cuestión que deseamos desinstalar.
2. Lo siguiente será seleccionar la app que queremos quitar y presionamos sobre el botón “**Desinstalar**” o “**Desactivar**”. Validamos esta decisión y esperamos a que el sistema termine. Cuando el procedimiento de desinstalación termine, presionamos sobre “**Ok**” y listo, el programa estará desinstalado.



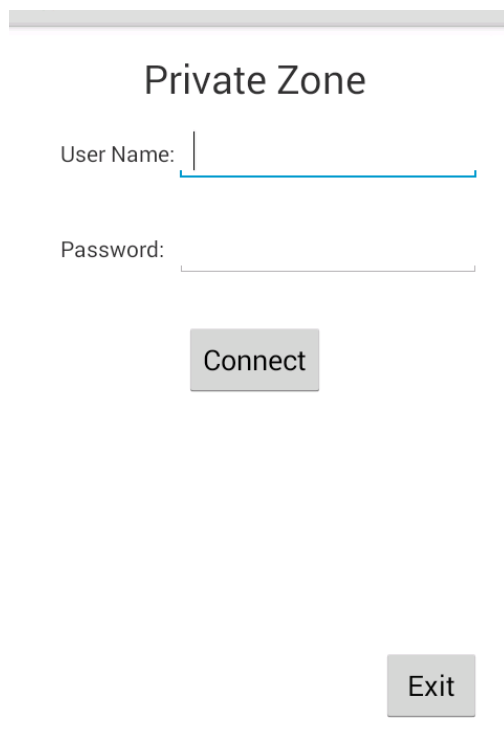
*Ilustración 32- Ventana diálogo sobre la desinstalación de una aplicación*

En el apartado de servidor y base de datos se procedería de la misma manera que en cualquier otra plataforma

## Manual de Usuario

### *Acceso privado a la aplicación*

En la siguiente pantalla, se presenta dos campos de texto. El primer campo representa el nombre de usuario y el segundo campo, la contraseña ligada a su cuenta.



*Ilustración 33- Pantalla de Inicio*

Una vez rellenados los campos, el usuario tendrá que clicar al botón Conectar para tener total acceso a los servicios. El servidor realizará la autenticación oportuna y trasladará automáticamente al usuario a la siguiente pantalla.

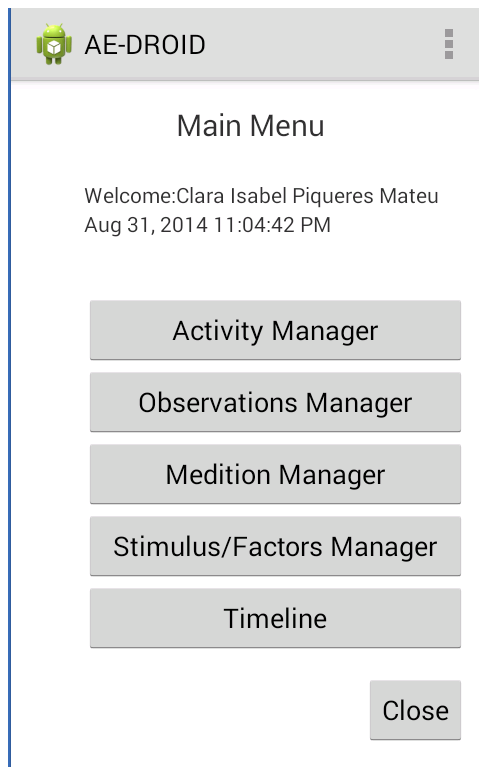
De lo contrario, si el proceso no se realizó bien, se le notificará en la pantalla un error de conexión de red, compruebe que los datos de acceso sean los correctos antes de llamar a su administrador de red habitual.

### *Navegación acceso común*

La pantalla principal se presenta un menú de navegación bastante intuitivo, el cual el usuario puede elegir las siguientes opciones:

- Gestor de Actividades: comprenden todas aquellas funciones relacionadas con la gestión de las actividades sobre las necesidades fisiológicas o pedagógicas.
- Gestor Observaciones: en esta sección, se podrá añadir, modificar o eliminar observaciones de cada medición realizada.
- Timeline: Muestra en varias secciones, diferentes datos organizados para facilitar el seguimiento de los alumnos

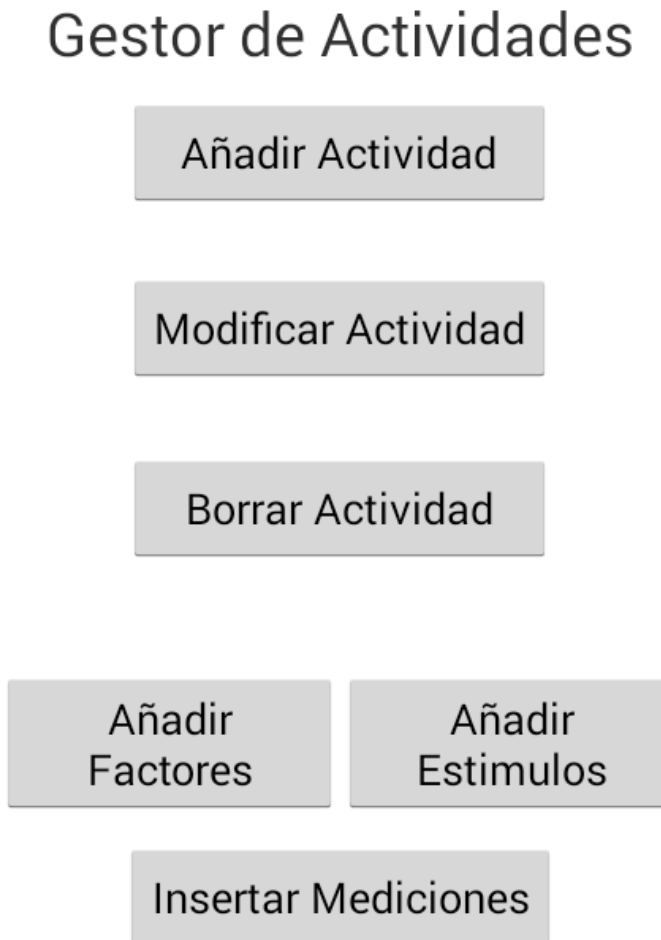
La siguiente imagen muestra el menú principal con las opciones explicados anteriormente:



*Ilustración 34- Menú Principal*

### *Gestor de Actividades*

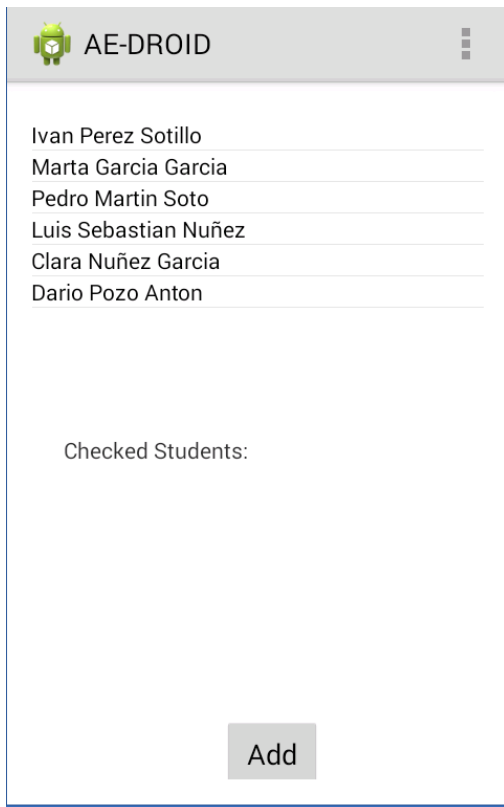
En esta pantalla se presenta un menú intuitivo donde el usuario podrá: dar de alta, modificar, eliminar actividades. Además de asociarlos a nuevos factores o estímulos e introducir mediciones en ellos. El idioma está en español para que se pueda observar la internacionalización del software.



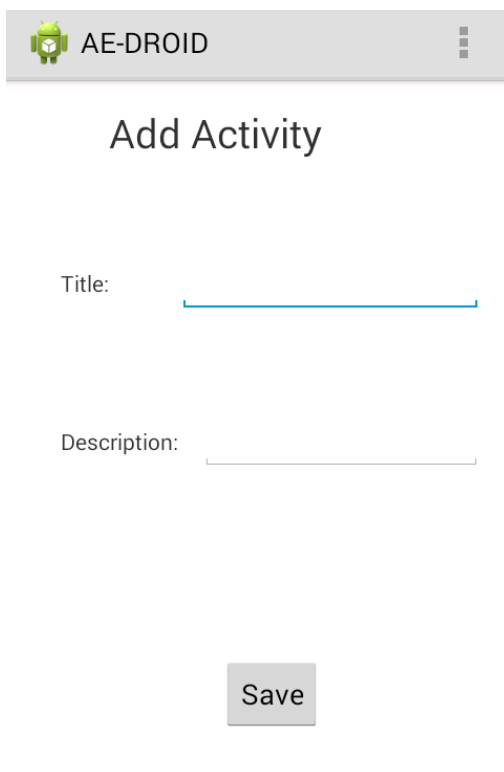
*Ilustración 35 - Pantalla Gestor Actividades (en español)*

### *Alta Actividad*

Para añadir una actividad, previamente hay que seleccionar que alumnos están involucrados en el, y posteriormente indicar un título nuevo y una breve descripción del mismo (opcional)



*Ilustración 36- Pantalla 1º Añadir Actividad*

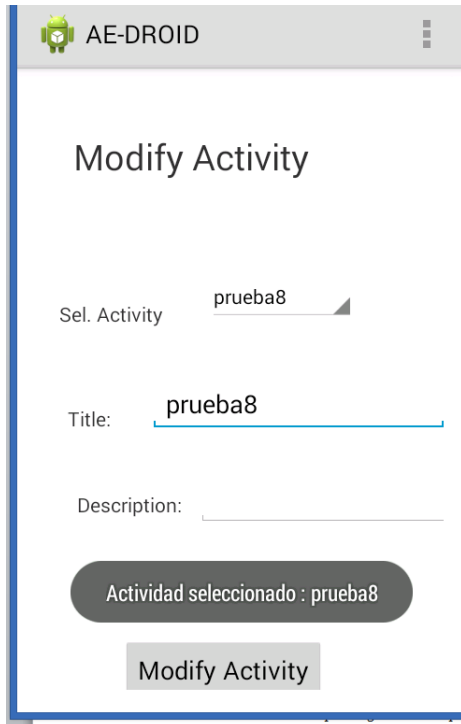


*Ilustración 37- Pantalla 2º Añadir Actividad*

***Modificar Actividad***

Para modificar una actividad, hay que seleccionar una actividad a través de la lista desplegable, como indica la siguiente figura:

Una vez seleccionado, se puede realizar la modificación escribiendo sobre los campos de texto y posteriormente aceptar los cambios, clicando el botón Modificar.

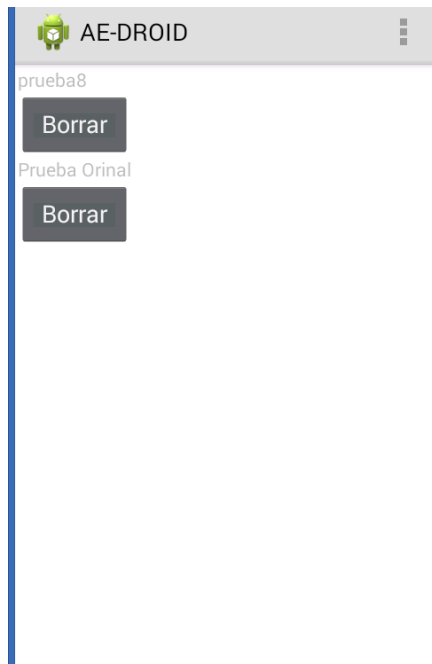


*Ilustración 38- Pantalla Modificación de Actividades*

### ***Borrar Actividad***

Para borrar una actividad, primero selecciona dicha actividad y posteriormente se acepta definitivamente la eliminación.





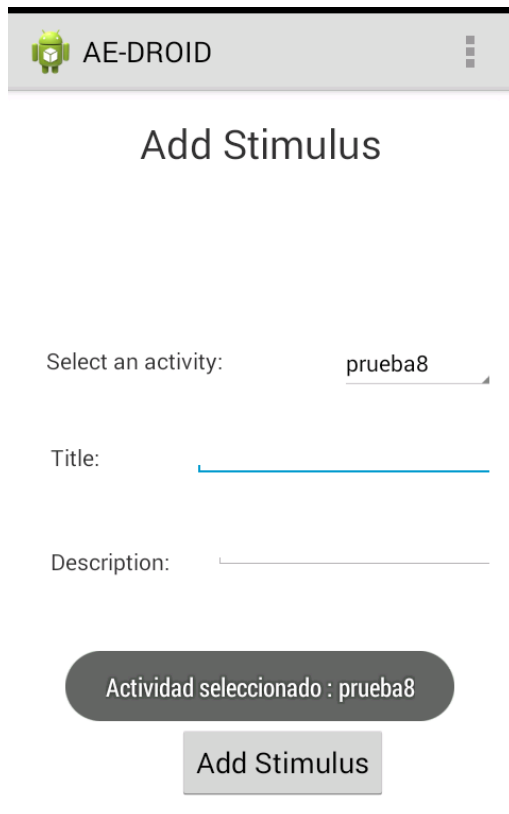
*Ilustración 39 Pantalla Borrar Actividad*

### *Añadir Estimulo*

Si desea añadir un estímulo se accede a través del menú de añadir actividades, clicar el botón Insertar Estimulo. Posteriormente se le pedirá que rellene varios campos de texto:

- Título: Este campo obligatorio corresponde con el título del estímulo en cuestión.
- Descripción: Este campo corresponde a la descripción del estímulo.

Una vez haya rellenado los campos correspondientes, si desea guardar los cambios, pulse el botón Insertar en la parte inferior de la pantalla. Se le notificará del estado de la operación al finalizar.



AE-DROID

## Add Stimulus

Select an activity: prueba8

Title: \_\_\_\_\_

Description: \_\_\_\_\_

Actividad seleccionado : prueba8

Add Stimulus

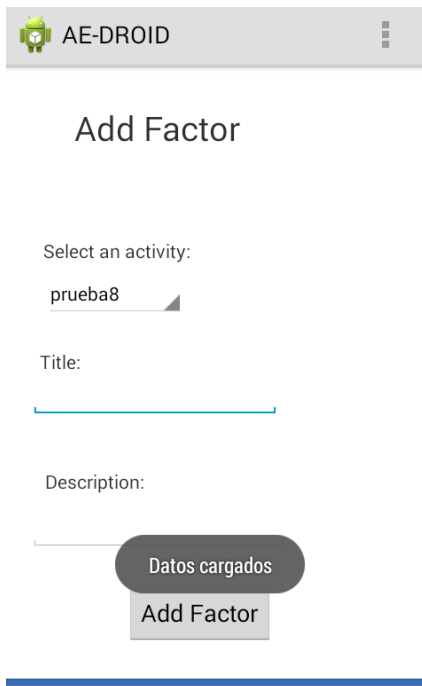
*Ilustración 40- Pantalla Añadir Estímulos*

### ***Añadir Factor***

Si desea añadir un factor se accede a través del menú de añadir actividades, clicar el botón Insertar Estimulo. Posteriormente se le pedirá que rellene varios campos de texto:

- **Título:** Este campo obligatorio corresponde con el título del factor en cuestión.
- **Descripción:** Este campo corresponde a la descripción del factor.

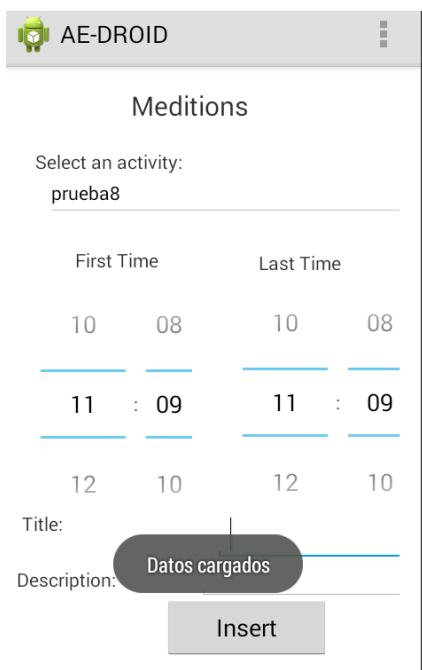
Una vez haya rellenado los campos correspondientes, si desea guardar los cambios, pulse el botón Insertar en la parte inferior de la pantalla. Se le notificará del estado de la operación al finalizar.



*Ilustración 41 - Pantalla Añadir Factores*

### *Insertar Mediciones*

Para insertar una medición, es necesario añadir un título previamente y marcar la hora de inicio y fin. También se puede añadir una breve descripción. Para finalizar se clica en el botón Insertar como indica la siguiente figura.

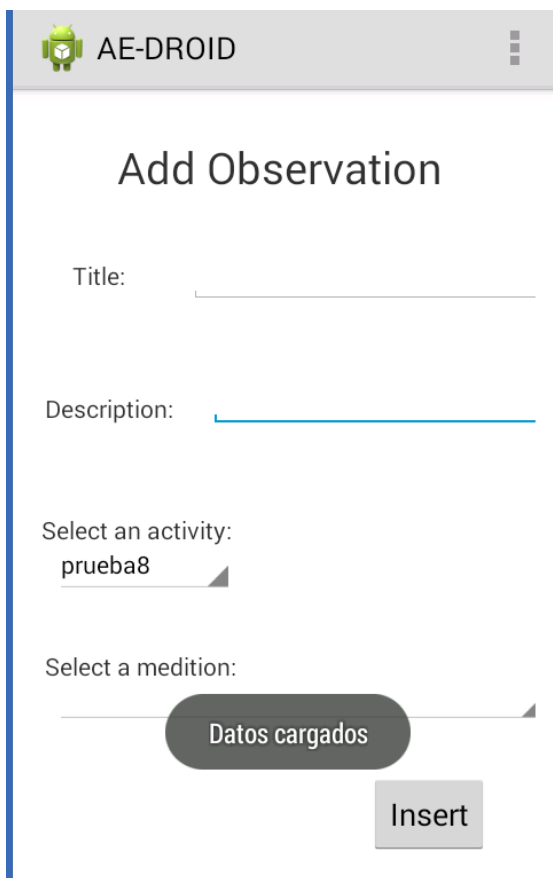


*Ilustración 42- Pantalla Añadir Medición*

### *Gestor de Observaciones*

### *Añadir Observaciones*

Para añadir una observación, se selecciona una actividad, después una medición y luego se rellenan los campos de texto título y descripción

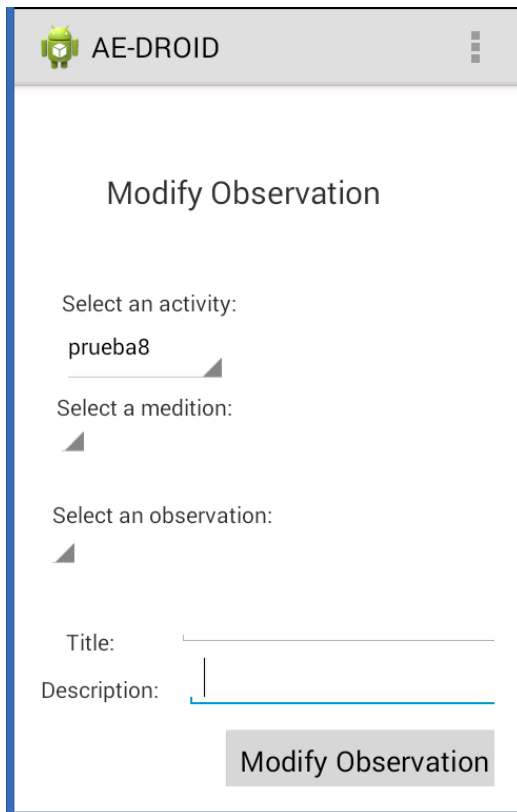


The screenshot shows the 'Add Observation' screen in the AE-DROID application. The screen has a grey header with the Android logo and 'AE-DROID' text. Below the header, the title 'Add Observation' is centered. There are four input fields: 'Title:' with an empty text box, 'Description:' with an empty text box, 'Select an activity:' with a dropdown menu showing 'prueba8', and 'Select a meditation:' with a dropdown menu showing 'Datos cargados'. At the bottom right, there is a grey 'Insert' button.

*Ilustración 43- Pantalla Añadir Observación*

### *Modificar Observaciones*

Para modificar una observación, primero se selecciona una actividad, luego una medición y posteriormente la observación deseada. Luego modificar el título y la descripción al texto deseado. Para aplicar los cambios, dar click al botón Modificar.



AE-DROID

## Modify Observation

Select an activity:  
prueba8

Select a meditation:  
▲

Select an observation:  
▲

Title: \_\_\_\_\_

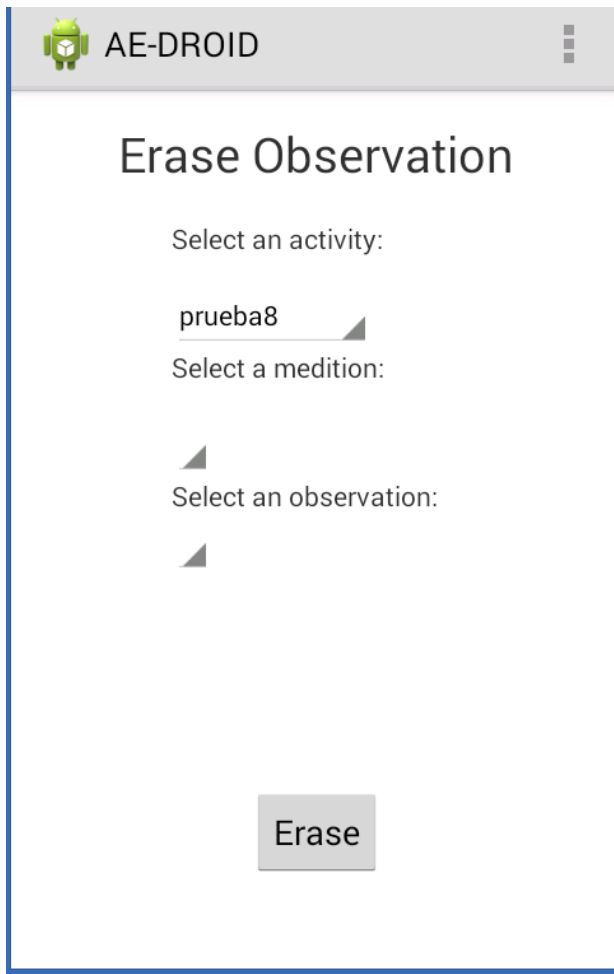
Description: \_\_\_\_\_

Modify Observation

*Ilustración 44- Pantalla del modificado de observación*

### ***Eliminar Observaciones***

Para eliminar una observación, solo basta con aplicar el botón Borrar de la fila deseada y aceptar los cambios.



*Ilustración 45- Pantalla Borrado de Observaciones*

### *Timeline*

En el timeline, se muestran varias pestañas dependiendo del tipo de ordenación se quiera mostrar: Temporal, por Estímulos o por Factores.

En cada uno de ellos se mostraran los alumnos relacionados con una determinada actividad ordenado por fecha o otras categorías.

Para recorrer el listado solamente deslice con el dedo verticalmente por la pantalla, desplegando aquellas categorías que se quieran.

## BIBLIOGRAFIA

A continuación se detallaran los siguientes recursos bibliográficos utilizados para el desarrollo de este documento:

- Pressman, INGIENERIA DEL SOFTWARE: Un enfoque práctico. Edición Séptima, McGraw Hill.
- Artículo Web, Desinstalación de aplicaciones en Android, Fecha de consulta el 27/08/14 en <http://www.androidexperto.com/aprender-android/aprende-eliminar-aplicaciones-preinstaladas-android/>
- Ruiz Francisco, Ingeniera del Software: Diseño de Software, Universidad de Cantabria  
<http://www.ctr.unican.es/asignaturas/is1/is1-t04-trans.pdf>
- Mario G. Piatinni, Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión, Ed. Ra-Ma
- Aguilera, Antonio. Introducción a Las Dificultades Del Aprendizaje. (2004). McGraw Hill
- Alumnado con necesidad específica de apoyo educativo. Ministerio de Educación  
<http://www.educacion.gob.es/educacion/sistema-educativo/educacion-inclusiva/necesidad-apoyo-educativo.html>

Apuntes de las asignaturas de Grado en Ingeniería Informática de Servicios y Aplicaciones:

- Proceso de Desarrollo del Software (2011/2012 ) de Jesús Álvarez.
- Gestión de Proyectos basados en las TI (2012/2013) de Francisco José González Cabrera.
- Plataformas Software Empresariales (2012/2013) de Aníbal Bregón
- Administración de Bases de Datos (2012/2013) de Miguel Ángel Martínez Prieto.
- Análisis de Requisitos (2013/2014) de Miguel Ángel Martínez Prieto.

## OTROS RECURSOS

- Eclipse:

[https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/luna/R/eclipse-java-luna-R-win32-x86\\_64.zip](https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/luna/R/eclipse-java-luna-R-win32-x86_64.zip)

- SQLyog Community Edition:

<http://goo.gl/4yMK5A>

- ProjectLibre:

<http://sourceforge.net/projects/projectlibre/>

- WhiteStarUML

<http://sourceforge.net/projects/whitestaruml/>