



# MejoraTuIngles

Trabajo de Fin de Carrera



# Introducción

Esta presentación se corresponde con el Trabajo de Fin de Carrera de los estudios de Telecomunicaciones, cursados en la Universidad Oberta de Catalunya.

El objetivo de este documento es mostrar, de forma global, el trabajo realizado durante la realización de este TFC. Con el fin de cumplir con los requerimientos de espacio, se ha realizado un esfuerzo de síntesis, mostrando únicamente los aspectos más relevantes del trabajo.

A modo de guía del documento, se han utilizado los hitos entregables más relevantes desde el punto de vista técnico para este TFC:

- Proyecto de TFC
- Estado del Arte
- Diseño e Implementación

# Proyecto de TFC: Objetivos

Los objetivos que se pretenden alcanzar pueden dividirse en dos ámbitos: objetivos técnicos y objetivos de gestión del proyecto. A continuación se presenta una enumeración de ambos.

## Objetivos Técnicos

- Conocer y manejar el concepto de e-learning.
- Aplicar la tecnología para el desarrollo de un prototipo de aplicación m-learning.
- Gestionar correctamente un desarrollo software relacionado con el ámbito de los sistemas e-learning

## Objetivos de Gestión

- Gestión de proyectos
- Presentación de información de forma clara y efectiva

Para alcanzar los objetivos propuestos, se creará **un prototipo de curso de inglés para dispositivos móviles**. El curso estará compuesto por:

- Unidades didácticas: Enfocadas al aprendizaje del idioma inglés.
- Exámenes: Para evaluar la comprensión de cada unidad didáctica. El sistema corregirá las respuestas del usuario y puntuará dicha unidad.
- Control de lecciones: El alumno podrá consultar las lecciones finalizadas y las notas de los exámenes.
- Ayuda: Para orientar al alumno sobre la utilización del curso.



En esta fase se realizó un estudio sobre el estado actual del e-learning a nivel global y los diferentes tipos enfoques existentes, para terminar profundizando en el m-learning que es la vertiente de e-learning que se desarrolla sobre plataformas móviles.

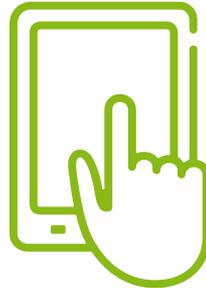
Tipos de E-learning, según el dominio temporal:

- **E-learning asíncrono:** Las actividades están disponibles en un servidor y son accedidas bajo demanda de los propios usuarios.
- **E-learning síncrono:** Necesita sincronización entre alumno y docente, simulando un aula tradicional a distancia. Si se graban las sesiones se convierten en asíncronas.



Tipos de E-learning, según la plataforma elegida:

- **B-learning:** Aprendizaje mixto entre enseñanza presencial y e-learning, utilizando en una clase presencial material multimedia.
- **Web Based Training:** El material es visualizado mediante un navegador web, incluyendo formato HTML, correos electrónicos, foros de debate, etc.
- **Computer Based Training:** Programas y tutoriales diseñados para una plataforma concreta.
- **M-Learning:** Se accede a los contenido mediante dispositivos móviles como teléfonos, tabletas, PDAs o Pocket PC.



## Ventajas

- *Sin necesidad de planificación previa*
- *Aprendizaje “outdoor”*
- *Acceso inmediato*

## Inconvenientes

- *Limitación por la duración de la batería*
- *Tamaño de la pantalla menor que en un PC*
- *Recursos del dispositivo limitados: memoria y microprocesador*

# Diseño e Implementación: Análisis

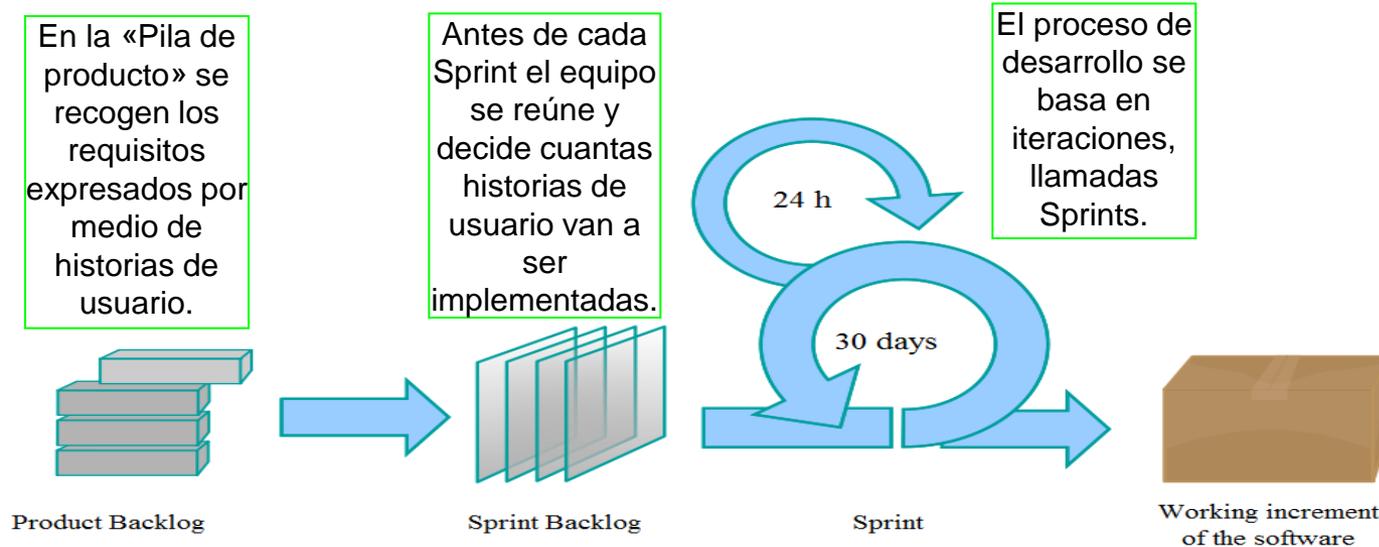
En la fase de análisis se realiza un estudio del problema desde el punto de vista del negocio. En primer lugar se realiza un análisis de los requisitos expresados por el cliente. Obtenidos estos requisitos se diseña y modela en UML un sistema capaz de dar respuesta a los mismos. Estos diseños servirán de input para la fase de implementación.

Las tareas que se llevan a cabo en esta fase son:

1. Presentación de la metodología SCRUM. Esta metodología es la elegida para la ejecución de este proyecto.
2. Estudio del problema y obtención de los requisitos funcionales de usuario.
3. Representación de los requisitos de usuario por medio de Historias de Usuario y diagrama de casos de uso.
4. Análisis UML del conjunto del sistema (diagrama de clases)
5. Análisis UML de las funcionalidades detectadas en las Historias de Usuario
6. Diseño de la interfaz gráfica de usuario
7. Diseño del plan de pruebas de usuario

SCRUM es una metodología enfocada a la gestión de desarrollos rápidos de software. Se ha elegido esta metodología por los siguientes motivos:

- Fácil de seguir
- Orientada a desarrollo iterativo e incremental
- Puede combinarse con técnicas de programación extrema.
- Se adaptan a las circunstancias de cada proyecto.



En la realización del proyecto se ha utilizado principalmente la captura de requisitos mediante historias de usuario y las ideas de desarrollo iterativo e incremental

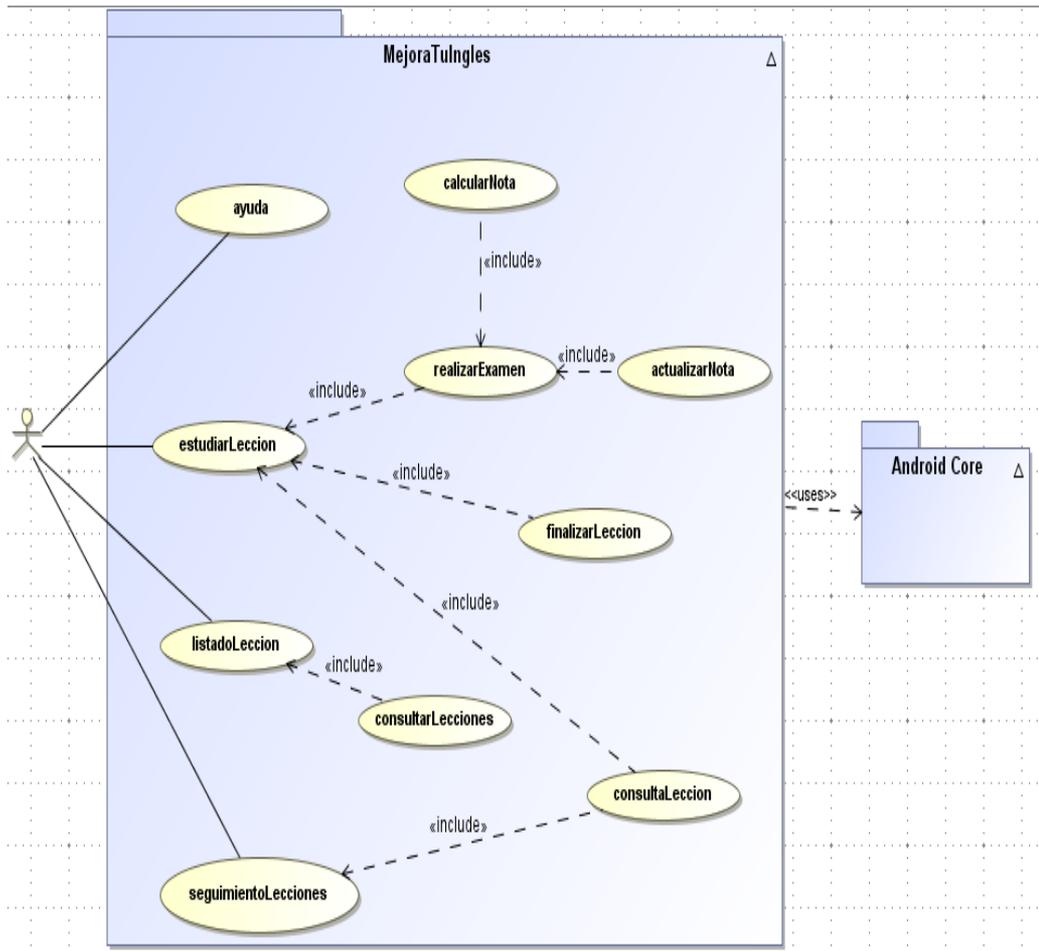
Una historia de usuario es la expresión, simple y coloquial, de una funcionalidad deseada por el usuario. En este proyecto se han registrado las siguientes historias de usuario:

- **Estudiar unidad didáctica:** Un usuario que estudia inglés quiere poder hacerlo aprovechando el tiempo que le dedica diariamente al transporte. Para ello quiere tener la posibilidad de estudiar temas de forma guiada en un dispositivo móvil.
- **Realizar examen:** Un usuario que ha concluido una unidad lectiva quiere comprobar el grado de asimilación de la misma, para ello querría tener la posibilidad de realizar un test que le ayude a evaluar los conocimientos adquiridos, indicándole los fallos que cometa.
- **Seguimiento de progresos:** El usuario hace uso del sistema cuando su tiempo se lo permite, este tiempo puede no seguir una pauta regular, teniendo como consecuencia que pueden pasar días entre una sesión de estudio y otra. Al usuario le gustaría poder marcar su nivel de avance, señalando de algún modo las unidades didácticas que ha completado.



# Diseño e Implementación: Casos de Uso

De las historias de usuario se han extraído los siguientes casos de uso:



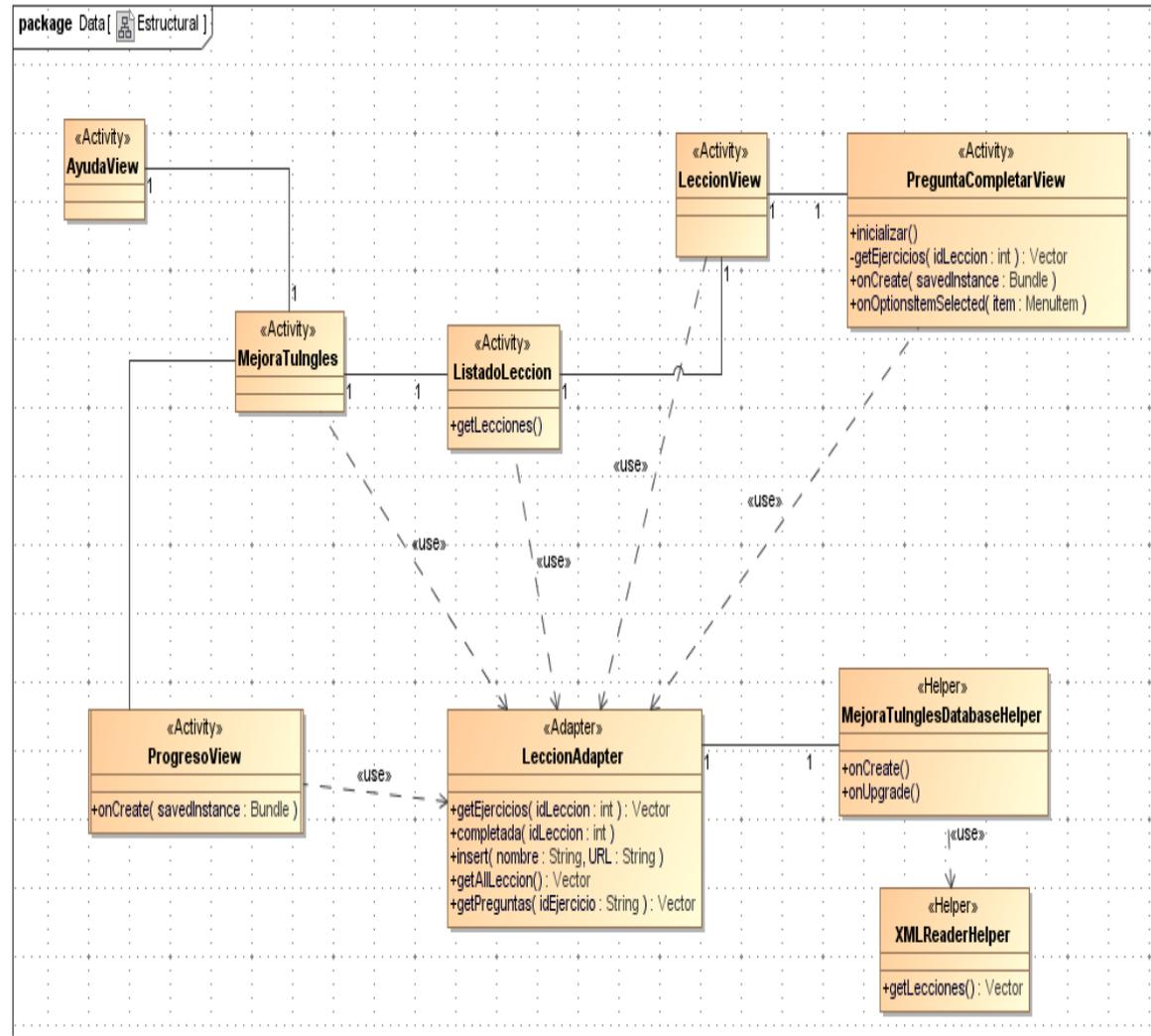
Esta presentación se centrará en los casos de uso más interesantes desde el punto de vista técnico.

# Diseño e Implementación: Diagrama de Clases

Identificada la funcionalidad, se propone un sistema que pueda dar respuesta a todos los problemas planteados.

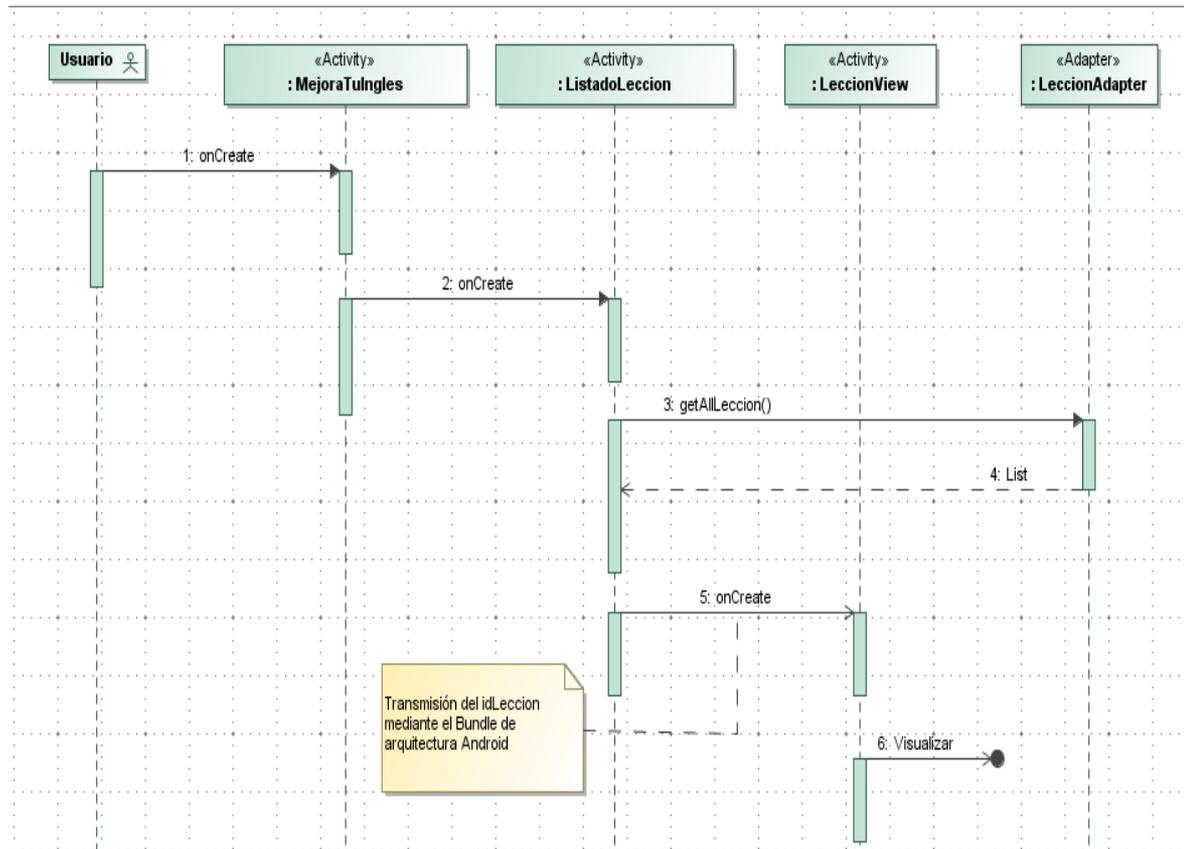
Para modelizar el sistema, utilizaremos distintos tipos de diagramas UML que representan diferentes aspectos del sistema.

El diagrama de clases permite representar las clases que componen el sistema y la relación existente entre ellas.



# Diseño e Implementación: Diagramas de secuencia

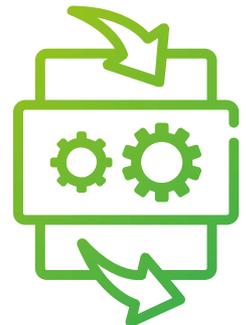
Para ilustrar las soluciones técnicas dadas se ha optado por la utilización de diagramas de secuencia, de esta forma se puede mostrar la interacción de los objetos intervinientes en el proceso ubicados en un marco temporal. Los diagramas muestran las llamadas, las respuestas y los parámetros que se intercambian los objetos. Todos los diagramas de solución técnica vienen incluidos en la memoria de este TFC. A modo de ejemplo podemos ver la secuencia de Estudio de Lección:



Las herramientas utilizadas para el desarrollo son las siguientes:

- **Eclipse** es un entorno de desarrollo, de propósito general, que admite varios lenguajes y puede ampliar sus prestaciones por medio de plug-ins.
- **Android SDK** es el kit de desarrollo estándar para Android que contiene:
  - Sistema Operativo Android.
  - SQLite : SGBD especialmente diseñado para disminuir el consumo de recursos.
  - AVD: Dispositivo virtual Android, incorporado al kit con fines de testing.

Ambas herramientas se integran mediante el plugin ADT, que permite utilizar las herramientas incorporadas en Android SDK desde Eclipse. La explicación detallada de la configuración del entorno puede consultarse en el documento «Anexo IV – Configuración del entorno de desarrollo».



# Diseño e Implementación: Interfaces de Usuario

El desarrollo de interfaces de usuario se basa en la creación de ficheros XML que describen la configuración que se visualizará en pantalla.

Cada actividad Android tiene asociado un fichero que describe su aspecto y que está compuesto por gestores de posicionamiento (**Layouts**) y por componentes a mostrar (textos, scrolls, imágenes)

```
package es.uoc.mejoratuingles.activities;

import android.os.Bundle;

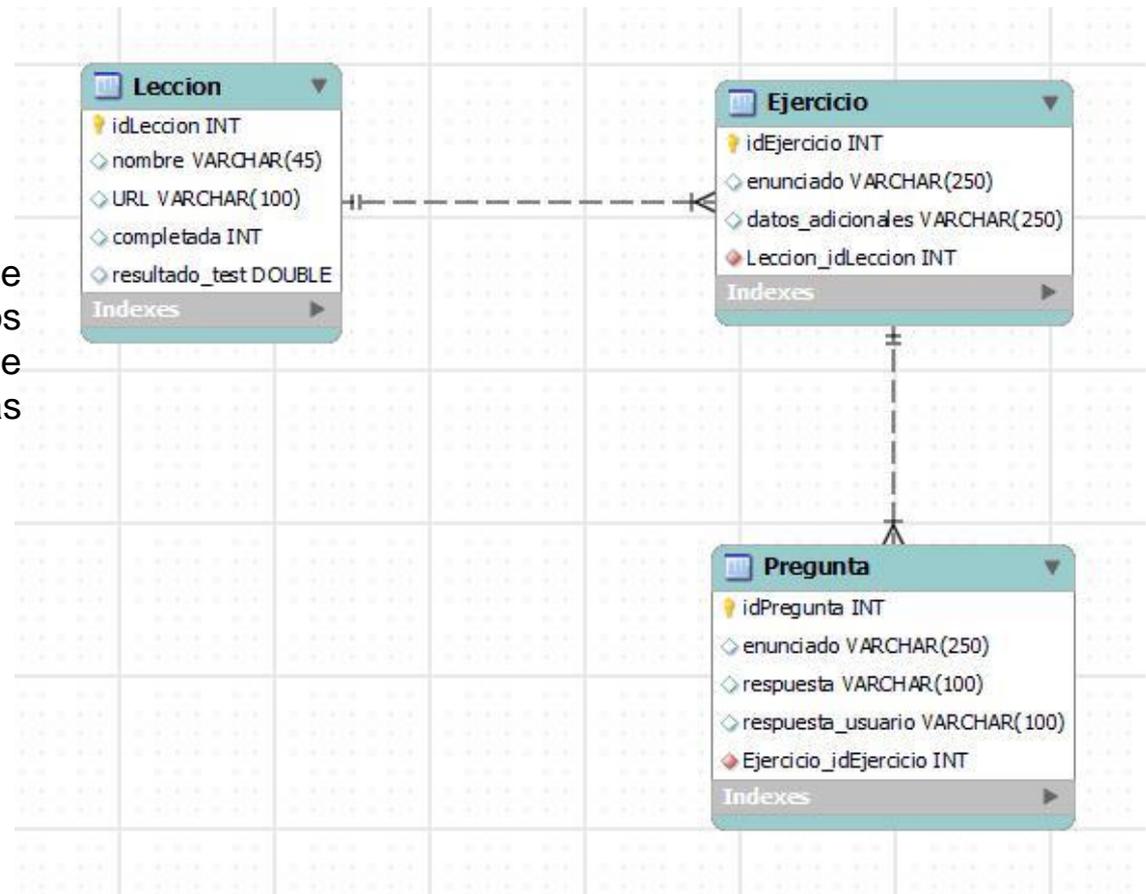
public class MejoraTuIngles extends Activity {
    // Establecimiento de Layout
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mejora_tu_ingles);
    }
    // Menú
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_mejora_tu_ingles, menu);
        return true;
    }
}
```

```
activity_mejora_tu_ingles.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <ImageView
        android:id="@+id/logotipo"
        android:src="@drawable/logotipo"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:contentDescription=""
    />
</LinearLayout>
```

Desde la clase Java de la actividad pueden recuperarse estos elementos y realizar acciones sobre ellos. Además, pueden asignarse **Listeners** para reaccionar ante eventos originados por el usuario.

**SQLite** es el **SGBD** incorporado en Android por defecto. Se ha utilizado en este TFC para proporcionar persistencia a la aplicación. El modelo de datos definido está pensado para dar respuesta a las necesidades detectadas en los casos de uso.

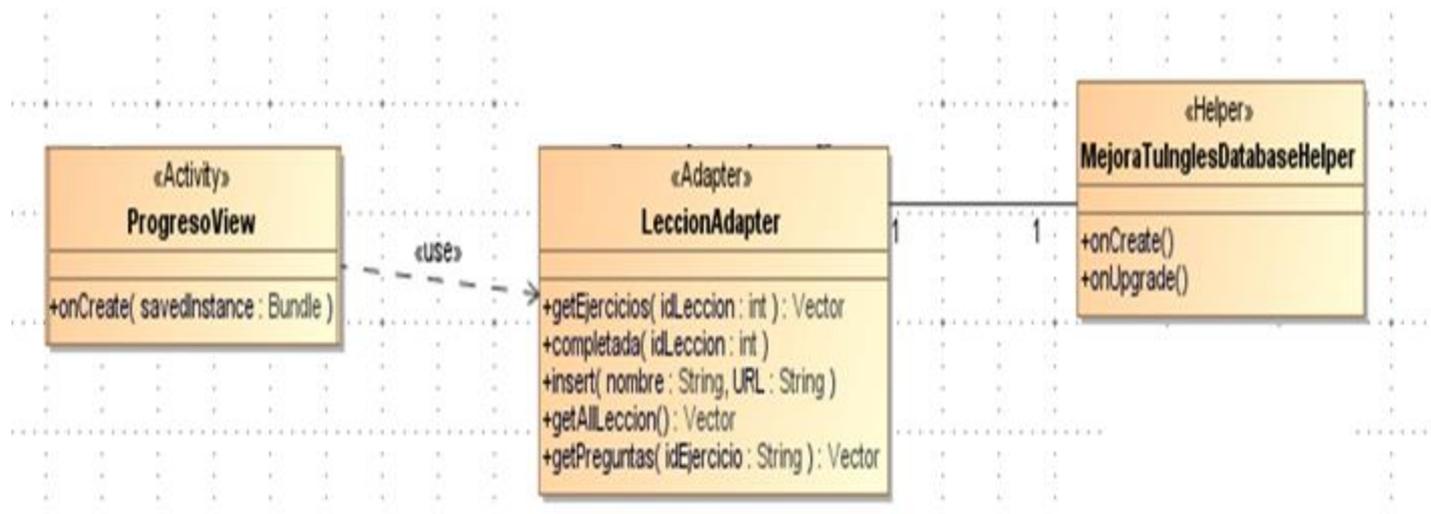
Para operar con BBDD, Android provee la clase **SQLiteOpenHelper**, debemos extender la funcionalidad de esta clase con el fin de adaptarla a las particularidades de nuestro negocio.



# Diseño e Implementación: Operaciones con BBDD

Con el fin de estructurar las operaciones con BBDD, se ha creado un sistema basado **Helpers** y **Adapters**. El esquema general consistiría en que cada entidad de BBDD contaría con una clase **Adapter** que implementaría las funcionalidades **CRUD** asociadas a esa entidad. Los **Adapters** se aislarían de los detalles de la conexión con BBDD mediante una clase **Helper**, que se encargaría de proveer esta conexión.

Siguiendo este patrón, cualquier actividad puede operar con BBDD aislándose de los detalles técnicos de la operación. La actividad únicamente deberá centrarse en el negocio.

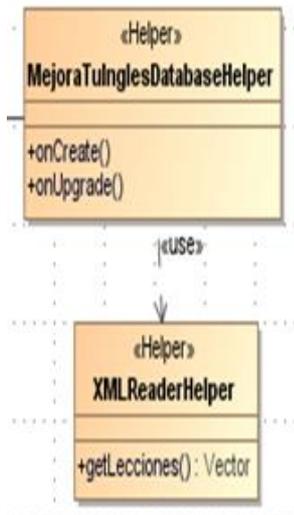


Esquema de operación de BBDD

# Diseño e Implementación: Gestión de XML

La carga inicial de lecciones, preguntas y respuestas se realiza mediante un fichero XML. Mediante este diseño es fácil añadir elementos a la aplicación sin modificar el código Java. Cuando se arranca la aplicación y se detecta que la versión de BBDD ha cambiado, automáticamente se lee el fichero XML de lecciones y se realiza una nueva carga, actualizando los contenidos.

Se ha creado la clase *XMLReaderHelper* para aislar a otras clase de los detalles de la lectura y tratamiento del fichero XML.



```

<leccion nombre="Present Simple" url="PresentSimple.html">
  <ejercicio>
    <enunciadoGeneral>
      <![CDATA[
        Complete the sentences with the following verbs in the correct form:
      ]]>
      <datosAdicionales>get - happen - look - lose - make - start - stay - try - work</datosAdicionales>
    </enunciadoGeneral>
    <pregunta>
      <enunciado>You ..... hard today</enunciado>
      <respuesta>are working</respuesta>
    </pregunta>
    <pregunta>
      <enunciado>I ..... for Christine. Do you know where she is?</enunciado>
      <respuesta>looking</respuesta>
    </pregunta>
  </ejercicio>
  <ejercicio>
    <enunciadoGeneral>
      <![CDATA[
        Put de verbs into the correct forms. Sometimes you need the negative. (I'm not doing etc.).
      ]]>
      <datosAdicionales></datosAdicionales>
    </enunciadoGeneral>
    <pregunta>
      <enunciado>Please don't make so much noise ..... (try) to work</enunciado>
      <respuesta>I'm trying</respuesta>
    </pregunta>
    <pregunta>
      <enunciado>Let's go out now. It ..... (rain) any more</enunciado>
      <respuesta>isn't raining</respuesta>
    </pregunta>
  </ejercicio>
</leccion>
  
```

Estructura XML de Leccion

# Diseño e Implementación: Prototipo

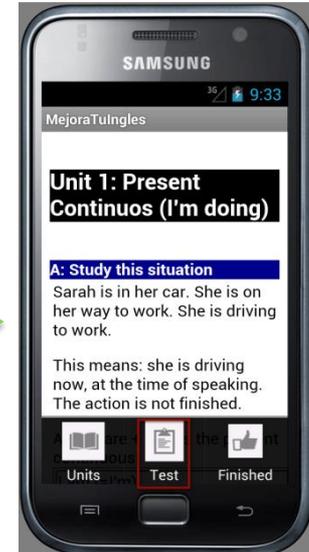
El proceso de estudio sería :



1 Seleccionamos "Units" desde el menú contextual.



2 La aplicación lista las lecciones disponibles y el usuario selecciona la que desea estudiar en ese momento.



3

El sistema muestra la lección seleccionada. Una vez terminada, se registra como tal pulsando la tecla "Finish" y el usuario puede realizar el test, pulsando el menú contextual "Test".



5

Con la tecla "Back" del menú contextual puede utilizar la navegación para completar todas las preguntas. El sistema mostrará un mensaje de aviso al finalizar el test.

El sistema responde mostrando el primer ejercicio de la unidad didáctica y la primera pregunta. Para responder el usuario utilizará el teclado virtual.

4



El proceso de consultar el progreso o la ayuda sería :



1 Seleccionamos "Progress" en el menú contextual.



2 El sistema muestra un cuadro referente al progreso en las diferentes unidades didácticas.



1 Seleccionamos "Help" en el menú contextual.



2 El sistema muestra una página de ayuda, en la que se explican las funcionalidades de la aplicación.