

UNIVERSITAT OBERTA DE CATALUNYA

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN  
ESPECIALIDAD DE TELEMÁTICA

TRABAJO FINAL DE CARRERA

---

# Aplicación Android para autoevaluar conocimientos

---

*Autor:*  
David CHINER BENJUYA

*Tutor:*  
Manel LLOPART VIDAL

14 de junio de 2013

## **Resumen**

En el presente trabajo de final de carrera se ha desarrollado una aplicación didáctica para que los estudiantes evalúen y amplíen sus conocimientos.

Dicha aplicación dispone de una interfaz gráfica desde donde cada estudiante puede consultar contenidos teóricos y realizar test de evaluación. En concreto, tras completar cada test, el estudiante repasa sus errores y recibe explicaciones sobre las respuestas correctas.

Para desplegar la aplicación se ha seleccionado la plataforma Android, un sistema operativo gratuito de enorme difusión en teléfonos móviles, tabletas e incluso *netbooks*. De este modo, la aplicación puede ser ejecutada en dispositivos que el estudiante lleve consigo.

### **Abstract**

The didactic application developed for this final thesis allows students to evaluate and expand their knowledge.

This application has a graphical interface where each student can read explanations and perform assessment test. Specifically, after completing each test, the student reviews his mistakes and gets explanations of the correct answers.

The application has been implemented for Android platform, a free operating system largely deployed in mobile phones, tablets and even netbooks. Thus, the application can be run on devices carried by the student.

## **Resum**

En el present treball de final de carrera s'ha desenvolupat un aplicació didàctica per a que els estudiants avaluïn i ampliïn els seus coneixements.

Aquest aplicació disposa d'una interfície gràfica des d'on cada estudiant pot consultar continguts teòrics i realitzar tests d'avaluació. Així, un cop completat cada test, l'estudiant repassa els seus errors i rep explicacions sobre les respostes correctes.

Per a desplegar l'aplicació s'ha seleccionat la plataforma Android, un sistema operatiu de gran difusió en telèfons mòbils, tabletas i fins i tot *netbooks*. D'aquesta manera, l'aplicació pot ser executada en dispositius que l'estudiant porti a sobre.

# Índice general

Índice de figuras	4
Índice de cuadros	5
<b>1. Introducción</b>	<b>6</b>
1.1. Palabras clave	6
1.2. Objetivos principales	7
1.3. Beneficios	7
1.4. Entregables	7
1.5. Motivación	7
1.6. Recursos	7
1.6.1. Curso desarrollo aplicaciones Android	7
1.6.2. Android - A programmer's guide	8
1.6.3. Stack Overflow	9
1.6.4. E-learning methodologies	9
1.6.5. The Not So Short Introduction to L <sup>A</sup> T <sub>E</sub> X	9
1.6.6. The Pomodoro Technique	10
<b>2. Estado del arte</b>	<b>11</b>
2.1. Titulaciones oficiales	12
2.1.1. Portales del estudiante	12
2.1.2. Titulaciones semipresenciales	12
2.1.3. Titulaciones distancia	12
2.2. Los MOOC	13
2.2.1. Udacity	13
2.2.2. Coursera	14
2.2.3. Miríada X	14
2.2.4. UOC CourseWare	14
2.3. Sistemas de Gestión de Aprendizaje	14
2.3.1. Moodle	14
2.3.2. Sakai	14
2.3.3. Piazza	15
2.4. Herramientas complementarias	15
2.4.1. YouTube	15
2.4.2. Facebook	15
2.4.3. Google Docs	15
2.4.4. Verbosus	16

2.5. Los dispositivos móviles . . . . .	16
2.5.1. iOS . . . . .	16
2.5.2. Android . . . . .	17
<b>3. Diseño . . . . .</b>	<b>18</b>
3.1. Actores . . . . .	18
3.2. Casos de uso . . . . .	18
3.2.1. Alta test . . . . .	18
3.2.2. Ejecución test . . . . .	19
3.3. Interfaces de usuario . . . . .	19
3.3.1. Interfaz alta test . . . . .	19
3.3.2. Interfaz resolución test . . . . .	19
3.4. Interfaces de comunicación . . . . .	20
3.4.1. Interfaz de comunicación web . . . . .	20
3.4.2. Interfaz de persistencia . . . . .	20
3.5. Interfaz software . . . . .	20
<b>4. Implementación con Android SDK . . . . .</b>	<b>21</b>
4.1. Entorno de desarrollo . . . . .	21
4.1.1. ADT Bundle . . . . .	21
4.1.2. Hoja de cálculo . . . . .	22
4.2. Diagrama de clases . . . . .	22
4.3. Manual de usuario . . . . .	23
4.3.1. Configuración de las series de test . . . . .	23
4.3.2. Ejecución de las series de test . . . . .	24
<b>5. Implementación con Android Apps . . . . .</b>	<b>28</b>
5.1. Implementación con doForms . . . . .	28
5.1.1. Elaboración de cuestionarios con doForms . . . . .	29
5.1.2. Ejecución de cuestionarios con doForms . . . . .	30
5.1.3. Consulta de resultados con doForms . . . . .	32
5.2. Implementación con Device Magic . . . . .	33
5.2.1. Elaboración de cuestionarios con Device Magic . . . . .	33
5.2.2. Ejecución de cuestionarios con Device Magic . . . . .	34
5.2.3. Consulta de resultados con Device Magic . . . . .	36
5.3. Comparativa . . . . .	36
<b>6. Conclusiones y líneas futuras . . . . .</b>	<b>38</b>
<b>A. Package com.uoc.dchiner_TFC . . . . .</b>	<b>39</b>
A.1. Class TfcMain . . . . .	41
A.1.1. Declaration . . . . .	41
A.1.2. Constructor summary . . . . .	41
A.1.3. Method summary . . . . .	41
A.1.4. Constructors . . . . .	41
A.1.5. Methods . . . . .	41
A.2. Class TfcPreferences . . . . .	42
A.2.1. Declaration . . . . .	42
A.2.2. Constructor summary . . . . .	42

A.2.3. Method summary . . . . .	42
A.2.4. Constructors . . . . .	42
A.2.5. Methods . . . . .	43
A.3. Class TfcPregunta . . . . .	45
A.3.1. Declaration . . . . .	45
A.3.2. Constructor summary . . . . .	45
A.3.3. Method summary . . . . .	46
A.3.4. Constructors . . . . .	46
A.3.5. Methods . . . . .	46
A.4. Class TfcResultados . . . . .	46
A.4.1. Declaration . . . . .	46
A.4.2. Constructor summary . . . . .	46
A.4.3. Method summary . . . . .	47
A.4.4. Constructors . . . . .	47
A.4.5. Methods . . . . .	47
A.5. Class TfcSelector . . . . .	47
A.5.1. Declaration . . . . .	47
A.5.2. Constructor summary . . . . .	47
A.5.3. Method summary . . . . .	48
A.5.4. Constructors . . . . .	48
A.5.5. Methods . . . . .	48
A.6. Class TfcSolucion . . . . .	48
A.6.1. Declaration . . . . .	48
A.6.2. Constructor summary . . . . .	48
A.6.3. Method summary . . . . .	48
A.6.4. Constructors . . . . .	49
A.6.5. Methods . . . . .	49
A.7. Class TfcTestDownloader . . . . .	49
A.7.1. Declaration . . . . .	49
A.7.2. Constructor summary . . . . .	49
A.7.3. Method summary . . . . .	49
A.7.4. Constructors . . . . .	50
A.7.5. Methods . . . . .	50
A.8. Class TfcWorkbook . . . . .	50
A.8.1. Declaration . . . . .	50
A.8.2. Constructor summary . . . . .	50
A.8.3. Method summary . . . . .	50
A.8.4. Constructors . . . . .	50
A.8.5. Methods . . . . .	51
<b>Bibliografía</b>	<b>52</b>
<b>Índice alfabético</b>	<b>53</b>

# Índice de figuras

4.1. Diagrama UML del proyecto. . . . .	23
4.2. Editor de hojas de cálculo Libre Office Calc. . . . .	24
4.3. Pantalla principal de la aplicación Android. . . . .	25
4.4. Mensaje de error de la aplicación Android. . . . .	25
4.5. Selección de la serie de test en la aplicación Android. . . . .	26
4.6. Opciones de un test en la aplicación Android. . . . .	26
4.7. Resultado de un test en la aplicación Android. . . . .	27
4.8. Resultados de una serie de test en la aplicación Android. . . . .	27
5.1. Nuevo formulario doForms. . . . .	29
5.2. Nuevo proyecto doForms. . . . .	29
5.3. Nuevo dispositivo móvil en doForms. . . . .	30
5.4. Aplicación móvil doForms. . . . .	30
5.5. Selección de formulario en doForms. . . . .	31
5.6. Formulario doForms. . . . .	31
5.7. Opciones de publicación en doForms. . . . .	32
5.8. Consulta de resultados en doForms. . . . .	32
5.9. Lista de formularios Device Magic. . . . .	33
5.10. Nuevo formulario Device Magic. . . . .	34
5.11. Condiciones lógicas de un formulario Device Magic. . . . .	34
5.12. Aplicación móvil Forms de Device Magic. . . . .	35
5.13. Opciones de un formulario de la aplicación Forms. . . . .	35
5.14. Resultados de un formulario de la aplicación Forms. . . . .	36
5.15. Consulta de resultados en una hoja de cálculo Google. . . . .	36



# Índice de cuadros

4.1. Columnas de cada hoja de cálculo .....	24
---	----

# Capítulo 1

## Introducción

### Contents

---

<b>1.1. Palabras clave</b> . . . . .	<b>6</b>
<b>1.2. Objetivos principales</b> . . . . .	<b>7</b>
<b>1.3. Beneficios</b> . . . . .	<b>7</b>
<b>1.4. Entregables</b> . . . . .	<b>7</b>
<b>1.5. Motivación</b> . . . . .	<b>7</b>
<b>1.6. Recursos</b> . . . . .	<b>7</b>
1.6.1. Curso desarrollo aplicaciones Android . . . . .	7
1.6.2. Android - A programmer's guide . . . . .	8
1.6.3. Stack Overflow . . . . .	9
1.6.4. E-learning methodologies . . . . .	9
1.6.5. The Not So Short Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	9
1.6.6. The Pomodoro Technique . . . . .	10

---

En los últimos años se han popularizado las plataformas educativas online como alternativa a los estudios presenciales. Por motivos económicos, geográficos o de simple comodidad, estudiantes de todo el mundo optan por seguir sus estudios a través de la red.

Por otro lado, el número de dispositivos móviles (teléfonos inteligentes, tabletas) ha superado con mucho al de los clásicos ordenadores personales. A este éxito no es ajeno el sistema gratuito Android desarrollado por Google.

Este trabajo final de ingeniería técnica de telecomunicaciones explora las posibilidades de los dispositivos móviles con sistema operativo Android para ejecutar test educativos de autoevaluación. Dichos test son previamente elaborados por un profesor y descargados a través de la red.

Para ello, se ha desarrollado una aplicación Android a medida pero también se han explorado las posibilidades de configuración de aplicaciones genéricas disponibles en la tienda [Google Play](#).

### 1.1. Palabras clave

E-learning, test, evaluación, tutorial, Android, móvil

## 1.2. Objetivos principales

Los principales objetivos de este trabajo final de carrera son los siguientes:

- Estructurar en forma de contenidos didácticos los conocimientos adquiridos durante la carrera.
- Crear un entorno atractivo para que futuros estudiantes consulten dichos contenidos y evalúen mediante test su dominio de los mismos.
- Desplegar el nuevo entorno como aplicación con interfaz táctil para plataforma Android.

## 1.3. Beneficios

Disponer de un entorno atractivo de aprendizaje en una plataforma para dispositivos móviles. Esto es, en una aplicación “de bolsillo” que los estudiantes puedan utilizar en cualquier lugar para ampliar y evaluar conocimientos.

## 1.4. Entregables

Al completar este trabajo final de carrera se han entregado los siguientes archivos:

- Documentación del proyecto (motivación, descripción, decisiones de diseño).
- Código de la aplicación.
- Programa de instalación de la aplicación.

## 1.5. Motivación

Este trabajo final de carrera permite estructurar los conocimientos adquiridos en las diferentes asignaturas y presentarlos en una aplicación didáctica para futuros estudiantes.

También supone un primer contacto con el desarrollo de aplicaciones para entorno Android.

## 1.6. Recursos

Para realizar el trabajo final de carrera se han consultado los recursos descritos a continuación.

√ En la <a href="#">Bibliografía</a> se detallan otros materiales complementarios utilizados en este proyecto.
--

### 1.6.1. Curso desarrollo aplicaciones Android

Videotutorial de desarrollo de aplicaciones Android repartido en varias lecciones:

1. Configurar entorno de desarrollo  
<http://www.youtube.com/watch?v=2zJD4ASpfW8>
2. Conocer las herramientas del SDK de Android  
<http://www.youtube.com/watch?v=i1ha4ULoVOw>

3. Android para desarrolladores de juegos  
[http://www.youtube.com/watch?v=P\\_A9IX-chCo](http://www.youtube.com/watch?v=P_A9IX-chCo)
4. Conocer las APIs de Android  
<http://www.youtube.com/watch?v=WleEwzQsi1k>
5. El ciclo de vida de una activity  
<http://www.youtube.com/watch?v=5sdgcHQMmrnk>
6. Eventos single y multi touch  
<http://www.youtube.com/watch?v=cD1gMRjK1U8>

### Resumen

El *Curso desarrollo aplicaciones Android* es un videotutorial publicado por Jesús Conde ([OutKast](#)) en su canal de YouTube [VideoTutoriales.com](#).

Las diferentes lecciones de este videotutorial detallan todas las etapas de creación de una aplicación Android, desde la configuración del entorno de desarrollo hasta la captura de eventos de pantalla.

Cada lección combina apuntes teóricos sobre Android con sencillos ejemplos de programación o configuración.

Al final del videotutorial se obtiene una visión muy completa del mundo de desarrollo de aplicaciones para dispositivos Android.

### Aspectos relevantes para el TFC

De todas las guías de consultadas para elaborar este proyecto, esta ha sido idónea para perder el miedo inicial y lanzarse a desarrollar el primer prototipo de aplicación Android.

Por su formato vídeo, evidentemente “entra por los ojos” pero sobre todo anima a repetir en el propio ordenador los pasos explicados por el profesor.

Para entrar en detalle y elaborar desarrollos más complejos, se han consultado los demás recursos sobre Android descritos esta sección.

### 1.6.2. Android - A programmer's guide

Libro de J.F. DiMarzio publicado por McGraw-Hill en 2008.

### Resumen

*Android - A programmer's guide* es una guía de desarrollo de aplicaciones Android dirigida a lectores con conocimientos de programación.

A pesar de su nivel avanzado, sigue un planteamiento de tutorial que empieza con la creación de una sencilla aplicación de ejemplo para profundizar después en los parámetros de configuración o las funciones disponibles.

### Aspectos relevantes para el TFC

Los primeros capítulos del libro aportan una visión muy clara del entorno desarrollo para Android y del esqueleto de cada aplicación: *Manifest*, recursos, *layouts*.

También aborda funcionalidades utilizadas en este proyecto final de carrera como el desarrollo de formularios o la persistencia de datos.

### 1.6.3. Stack Overflow

Comunidad online para exponer y resolver dudas sobre programación:

<http://stackoverflow.com>

#### Resumen

*Stack Overflow* es una de las comunidades de la red *Stack Exchange* dedicadas a resolver dudas sobre temas específicos.

En concreto, en la comunidad online *Stack Overflow* se atienden dudas sobre software. Desde la configuración del entorno de desarrollo hasta la ejecución, pasando por la codificación de los programas.

La calidad de las respuestas se pondera mediante un sistema de puntuación que sitúa en primer lugar a las más votadas.

#### Aspectos relevantes para el TFC

Para resolver dudas sobre este proyecto han sido constantes las consultas mediante el buscador Google.

El número de respuestas encontradas en *Stack Overflow* ha sido excepcionalmente alto. No sólo al buscar soluciones a errores de programación sino también otros aspectos más específicos como la configuración del emulador de Android.

### 1.6.4. E-learning methodologies

Guía elaborada en 2011 por el *FAO Trust Fund Project* y disponible para descarga en:

<http://www.fao.org/docrep/015/i2516e/i2516e.pdf>

#### Resumen

El documento *E-learning methodologies* de la [Food and Agriculture Organization](#) (FAO) es una guía de buenas prácticas para la elaboración de cursos online.

Sigue un planteamiento genérico que empieza describiendo los beneficios del E-learning, explica cómo abordar un proyecto didáctico, y finalmente analiza las diferentes modalidades de difundir contenidos, comunicarse con los alumnos y evaluar conocimientos.

Para facilitar decisiones de implementación, incluye también una comparativa de las diferentes plataformas open-source de E-learning.

#### Aspectos relevantes para el TFC

La aplicación de E-learning desarrollada en este trabajo final de carrera es utilizada individualmente para consultar y autoevaluar conocimientos. Es decir, no permite seguir un curso completo ni facilita la comunicación entre profesores y alumnos.

No obstante, para que la aplicación sea lo más atractiva posible y estimule el aprendizaje son necesarias nociones generales de E-learning como las explicadas en el documento *E-learning methodologies*.

Además, dicho documento aborda aspectos utilizados en este proyecto final de carrera como la elaboración de cuestionarios de evaluación.

### 1.6.5. The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X

Tutorial de Tobias Oetiker disponible para descarga en:

<http://tobi.oetiker.ch/lshort/lshort.pdf>

### Resumen

El sistema de edición  $\text{\LaTeX}$  es complicado y áspero en un primer contacto pero a la larga acelera mucho la creación de documentos bien formateados.

A diferencia de los editores WYSIWYG, no da sorpresas de última hora con cambios de formato inesperados. Mediante el uso de plantillas, permite automatizar decisiones de diseño como empezar los nuevos capítulos en página impar. Además, facilita mucho la generación de tablas de contenido e índices temáticos.

*The Not So Short Introduction to  $\text{\LaTeX}$*  es un clásico en los manuales sobre el sistema de edición  $\text{\LaTeX}$ . Aunque su uso ha sido sustituido en gran parte por las búsquedas en Google (para resolver dudas puntuales), sigue siendo una referencia muy útil para iniciarse en el sistema de edición  $\text{\LaTeX}$  y abordar modos avanzados de formato.

### Aspectos relevantes para el TFC

Para redactar la memoria del presente proyecto se ha utilizado el sistema de edición  $\text{\LaTeX}$ . En concreto, se ha aplicado la plantilla *report* con algunas modificaciones para mejorar el aspecto de los enlaces o las tablas de contenido.

En este contexto, *The Not So Short Introduction to  $\text{\LaTeX}$*  es de especial utilidad por ser precisamente una guía muy orientada a la redacción de tesis o artículos científicos.

#### 1.6.6. The Pomodoro Technique

Guía de Francesco Cirillo disponible para descarga en:

<http://www.pomodortechnique.com/download/pdf/ThePomodoroTechnique.v1-3.pdf>

### Resumen

En el día a día de un proyecto se debe luchar con problemas como la famosa *procastination* (postergación) o las interrupciones (voluntarias o indeseadas).

Para combatir estos males, Francesco Cirillo elaboró una técnica de gestión de tiempo a la que denominó *pomodoro* por la forma de los relojes de cocina utilizados.

Así, un pomodoro es una unidad de indivisible de 25 minutos dedicados a una sola tarea. Durante un pomodoro no se puede atender el correo, responder a llamadas de teléfono o cambiar de tarea. Si una tarea se completa antes de concluir un pomodoro, debe dedicarse el resto del tiempo a repasar lo realizado.

### Aspectos relevantes para el TFC

La técnica pomodoro está especialmente indicada para favorecer la concentración cuando se trabaja desde casa. Por ejemplo, en un proyecto final de carrera.

Además, facilita mucho el cálculo del tiempo invertido en cada tarea. Basta con multiplicar por 25 el número de pomodoros registrados.

# Capítulo 2

## Estado del arte

### Contents

---

<b>2.1. Titulaciones oficiales</b>	<b>12</b>
2.1.1. Portales del estudiante	12
2.1.2. Titulaciones semipresenciales	12
2.1.3. Titulaciones distancia	12
<b>2.2. Los MOOC</b>	<b>13</b>
2.2.1. Udacity	13
2.2.2. Coursera	14
2.2.3. Miríada X	14
2.2.4. UOC CourseWare	14
<b>2.3. Sistemas de Gestión de Aprendizaje</b>	<b>14</b>
2.3.1. Moodle	14
2.3.2. Sakai	14
2.3.3. Piazza	15
<b>2.4. Herramientas complementarias</b>	<b>15</b>
2.4.1. YouTube	15
2.4.2. Facebook	15
2.4.3. Google Docs	15
2.4.4. Verbosus	16
<b>2.5. Los dispositivos móviles</b>	<b>16</b>
2.5.1. iOS	16
2.5.2. Android	17

---

Por *E-learning* o *aprendizaje electrónico* entendemos la educación a distancia mediante herramientas electrónicas.

Como en los cursos presenciales, en un curso en modalidad de E-learning el estudiante sigue las explicaciones de los profesores, consulta los materiales, asiste a tutorías y finalmente se somete a una prueba de evaluación.

Por tanto, este modo de estudio no es una simple consulta autodidacta de tutoriales o foros. Ni tampoco una implementación digital de los antiguos estudios a distancia o “por correspondencia”. En el E-learning, las herramientas telemáticas simulan la presencialidad y potencian la comunicación, tanto entre alumnos como entre profesor y alumno.

## 2.1. Titulaciones oficiales

En los últimos años, muchos centros educativos han querido aprovechar las ventajas de la enseñanza online para impartir titulaciones oficiales.

Por un lado, centros que imparten enseñanzas presenciales han ofrecido sus propios canales de E-learning. Por el otro, han aparecido centros que sólo imparten titulaciones de forma virtual.

En este contexto, se pueden distinguir diferentes modalidades de E-learning para impartir titulaciones oficiales.

### 2.1.1. Portales del estudiante

Los centros educativos han aprovechado el E-learning para conectar mejor con sus estudiantes. Típicamente, han creado portales web donde los estudiantes pueden interactuar con los profesores, consultar materiales de clase o enviar sus prácticas.

Por ejemplo, los estudiantes de la [Universitat Politècnica de Catalunya](#) (UPC) disponen del [portal Àgora](#) para consultar materiales e interactuar con la comunidad educativa.

En la medida en que facilitan el aprendizaje, dichos portales pueden considerarse herramientas de E-learning. En realidad, han supuesto un punto de partida para la implementación de cursos semipresenciales o a distancia.

### 2.1.2. Titulaciones semipresenciales

Por enseñanza semipresencial entendemos una modalidad que combina las clases presenciales con las virtuales. De este modo, el estudiante puede seguir a distancia los cursos de contenido más teórico. En cambio, para otros de tipo más aplicado debe asistir al centro educativo.

Varios centros presenciales han utilizado el E-learning para ofrecer en modo semipresencial titulaciones que ya ofrecían en modo presencial.

Por ejemplo, la [Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa](#) ofrece en esta modalidad su máster en ingeniería industrial.

### 2.1.3. Titulaciones distancia

Finalmente, se encuentran las titulaciones puramente *virtuales*. Se trata de estudios que de principio a fin se imparten en modo telemático. Dichas titulaciones sólo requieren la presencia física del estudiante para las pruebas de evaluación.

Por ejemplo, en la [Universitat Autònoma de Barcelona](#) (UAB) el mismo grado de estadística puede estudiarse en las modalidades presencial o virtual.

Más destacado es el caso de la [Universitat Oberta de Catalunya](#) (UOC) como modelo de universidad cien por cien telemática que ofrece titulaciones con la misma validez oficial que cualquier otra universidad.

A pesar de su entidad independiente como universidad oficial, la UOC mantiene una enriquecedora simbiosis con las universidades presenciales. Por ejemplo, en la página personal de [Jordi Torres](#) se encuentran materiales que este profesor ha elaborado para la UOC pero que también recomienda en la bibliografía de las asignaturas que imparte presencialmente en la UPC.

Además, la UOC ha establecido un convenio para que los grados que imparte se reconozcan también como *bàtxelor* de la [Universitat de Andorra](#) (UdA).

Destaquemos por último que el [Institut Obert de Catalunya](#) (IOC) ofrece también en modalidad virtual titulaciones oficiales de educación secundaria y formación profesional.



## 2.2. Los MOOC

Los MOOC (Massive Online Open Course) –en español COMA (Cursos Online Masivos Abiertos)– son la última revolución en el ámbito educativo, especialmente universitario.

Se trata de plataformas educativas que imparten cursos gratuitos por Internet. El único requisito para acceder es disponer de un ordenador conectado a la red. Al estudiante no se le exige matrícula ni prueba de acceso, aunque según el contenido del curso se recomienden conocimientos previos.

Los cursos son impartidos por profesores con experiencia, a veces de prestigio, y seguidos por alumnos de todo el mundo.

En general, las pruebas de evaluación se realizan exclusivamente a distancia y los alumnos que las superan obtienen un certificado de asistencia sin validez oficial. Además, las universidades que ofrecen MOOC no parecen muy dispuestas a convalidar dichos certificado por los créditos “de pago” que imparten a sus estudiantes presenciales.

Entre las ventajas de los MOOC destacan obviamente su carácter gratuito. Algo especialmente importante para personas sin posibilidad de pagar una matrícula universitaria.

Respecto a los inconvenientes, la rectora de la UOC Imma Turbella apunta lo siguiente en declaraciones a [La Vanguardia](#):

Son cursos de autoaprendizaje, cursos que se ofrecen independientemente del contexto de las personas, puesto que pueden tener hasta un millón de estudiantes a la vez, lo cual comporta una ratio de abandono altísima (cerca del 95 %) porque no hay acompañamiento. Existe el riesgo de que se considere la educación en línea de segunda categoría.

El problema de la alta ratio de abandono también es señalado por Chris Dede, profesor de tecnologías de la educación en Harvard. Así, en declaraciones a [Nature](#) explica que para perseverar en un curso los estudiantes necesitan apoyo tanto de sus compañeros como de sus profesores. Y que por ahora, las plataformas MOOC no han hecho lo suficiente para garantizar este apoyo. En realidad han dejado la resolución de este problema a los estudiantes:

They're just kind of hoping that people will figure out from the bottom up how to support each other.

Entre los MOOC más populares, destacan por ejemplo [Udacity](#), [Coursera](#) y [Miriada X](#).

### 2.2.1. Udacity

En 2011, un profesor de Stanford llamado Sebastian Thrun impartió un curso online sobre inteligencia artificial.

El curso fue seguido por 160.000 estudiantes de los que la mitad completaron el curso y obtuvieron el certificado de asistencia. El profesor Thrun concluyó que ya nunca podría volver a sus clases presenciales de 20 alumnos y decidió lanzar la plataforma Udacity de educación online.

Actualmente Udacity ofrece de un completo catálogo de cursos sobre gestión, informática, física o matemáticas. Dispone además de convenios con una red de centros presenciales –*Authorized Test Center*– donde realizar exámenes oficiales.

Finalmente, Udacity cuenta con un modelo de negocio. Aunque seguir sus cursos es gratis, cobra a las empresas que contratan a alumnos mediante la bolsa de trabajo del centro.

### 2.2.2. Coursera

Aunque más reciente que Udacity, Coursera es una de las plataformas de MOOC más populares. Cuenta con 2.7 millones de alumnos y ofrece cursos de 33 universidades de los cuatro continentes.

Precisamente parte de su éxito se debe al prestigio de las universidades que ofrecen sus cursos. Desde Stanford hasta la École Polytechnique pasando por la española IE Business School.

Además, Coursera se ha sumado al [American Council on Education's College Credit Recommendation Service](#) (CREDIT). Este proyecto ayuda a estudiantes con pocos recursos para que las universidades reconozcan los créditos que han obtenido al margen del sistema educativo tradicional.

### 2.2.3. Miríada X

En el mundo hispanohablante, destaca Miríada X, una plataforma que agrupa a 1.242 universidades de 23 países de Iberoamérica.

Como en el caso de Coursera, en Miríada X se entregan certificados de asistencia a los estudiantes que completan un curso. No obstante, también afirman estar en contacto con los centros educativos para que dichos certificados tengan validez oficial en el futuro.

### 2.2.4. UOC CourseWare

Aunque no participa en plataformas MOOC, la UOC comparte sus contenidos educativos mediante el [UOC OpenCourseWare](#). Desde este sitio web podemos descargar por ejemplo los famosos *materials* de todo tipo de asignaturas.

Por otro lado, desde el [Repositorio institucional O2](#) pueden consultarse tanto artículos científicos como los trabajos de final de carrera de sus estudiantes.

## 2.3. Sistemas de Gestión de Aprendizaje

Además de infraestructura de red y equipos informáticos, para impartir cursos en modalidad de E-learning es necesario un *Learning Management System* (LMS) o Sistema de Gestión de Aprendizaje.

Dicho sistema proporciona el software necesario para impartir los cursos online e interactuar con los alumnos.

### 2.3.1. Moodle

[Moodle](#) es una plataforma de código abierto para desplegar entornos educativos. Se trata de un sistema altamente modular y escalable, inicialmente desarrollado por Martin Dougiamas como tesis doctoral sobre código abierto para el aprendizaje.

Moodle es utilizado por universidades virtuales como la [Open University](#), pero también por universidades e institutos presenciales de todo el mundo para impartir cursos o simplemente crear portales web.

Moodle dispone de una comunidad de usuarios y empresas especializadas en despliegues a medida.

### 2.3.2. Sakai

Aunque Moodle es el sistema de gestión de aprendizaje más utilizado, existen alternativas como [Sakai](#).

Inicialmente desarrollado por la Universidad de Michigan y la Universidad de Indiana, el proyecto Sakai impulsa el desarrollo de plataformas educativas de código abierto.

Actualmente, en la [Sakai Foundation](#) participan universidades de todo el mundo.

### 2.3.3. Piazza

La funcionalidad de plataformas LMS como Moodle o Sakai puede extenderse con herramientas como [Piazza](#). En concreto, Piazza sustituye funcionalidades nativas de dichas plataformas para agilizar la creación de aulas virtuales y la interacción con los alumnos.

## 2.4. Herramientas complementarias

Para impartir cursos online, también se aprovechan herramientas genéricas como las descritas a continuación.

### 2.4.1. YouTube

En los MOOC, las clases se imparten en forma de vídeos que los alumnos siguen de forma no síncrona. Lo más habitual, es que al principio de cada semana el profesor publique los vídeos de una lección y los alumnos tengan de plazo hasta la semana siguiente para seguirlos y responder a las preguntas.

En este contexto, [YouTube](#) es el espacio habitual para compartir dichos vídeos.

Algunos MOOC como Coursera publican los vídeos en sus propios servidores. No obstante, al acabar cada curso los comparten también en su propio canal de YouTube para ponerlos a disposición de las personas que en su momento no se matricularon.

Otros MOOC como Miriada X publican los vídeos directamente en YouTube y los incrustan en su portal web.

### 2.4.2. Facebook

Las redes sociales han sustituido a las antiguas “cafeterías universitarias” donde comentar las clases, compartir materiales o simplemente hacer vida social.

Las plataformas MOOC suelen disponer de sus propios foros de estudiantes. En el caso de Coursera, también se intenta sacar partido a los hangouts de Google+, un tipo de videoconferencias colaborativas donde los estudiantes comentan simultáneamente aspectos relevantes del curso.

No obstante, [Facebook](#) es el canal más utilizado por los estudiantes para crear comunidades al margen de la institución educativa. Así, al principio del curso, uno de los temas en los foros *oficiales* es precisamente la creación de grupos en Facebook.

Al ser la red social más popular, acaba siendo el lugar natural donde debatir con estudiantes que siguen un mismo curso.

En los grupos de Facebook, los estudiantes comparten materiales del curso, intercambian impresiones o resuelven dudas.

### 2.4.3. Google Docs

Google Docs es el entorno de edición para los archivos almacenados en [Google Drive](#). Como sistema de publicación en la nube, es una herramienta idónea para editar y compartir documentación. Así, gracias a Google Docs:

- Los profesores publican los materiales de clase.
- Los estudiantes elaboran documentos colaborativos como resúmenes o prácticas.

Entre los formatos soportados por Google Docs, destaquemos los Google Forms, unos formularios en línea especialmente útiles para encuestas de satisfacción o test de autoevaluación.

#### 2.4.4. Verbosus

Además de los clásicos entornos para ordenador personal, el sistema de edición científica L<sup>A</sup>T<sub>E</sub>X cuenta con editores en línea como [LateX Labs](#) o [Verbosus](#).

Dichos editores permiten compilar el código en remoto y descargar el PDF resultante. Los posibles errores de compilación pueden analizarse en el archivo de log.

Verbosus cuenta además con la aplicación para dispositivo móvil [VerbTex](#) que permite por ejemplo redactar estas mismas líneas desde un vagón de metro.

## 2.5. Los dispositivos móviles

El dispositivo habitual para seguir un curso en modalidad E-learning es un ordenador personal (portátil o de sobremesa) ubicado en el hogar del alumno. Este tipo de dispositivo tiene ventajas obvias:

- Por el tamaño de la pantalla, permite consultar con claridad los materiales y vídeos.
- Por su capacidad de procesamiento, permite trabajar con agilidad y cambiar rápidamente entre tareas.
- Por su sistema operativo, soporta software sofisticado.
- Por su teclado físico, permite escribir cómodamente.

Por ello, de los ordenadores personales suele decirse que son el entorno natural de edición de contenidos, mientras que los dispositivos móviles quedarían relegados a tareas de consulta.

No obstante, el éxito experimentado en los últimos años por los dispositivos móviles tiende a borrar un poco esta diferencia en el uso.

De entrada, la mejora de las conexiones de banda ancha ha facilitado la navegación desde este tipo de dispositivos. Hasta el punto de que el número de accesos a Internet desde dispositivo móvil ya supera al de accesos desde ordenador personal.

Es más, los dispositivos móviles ya no son necesariamente una herramienta para consultas rápidas fuera del hogar. El usuario sentado en el sofá de su casa con una tableta se ha convertido en uno de los principales actores de de Internet.

Y es que las tabletas y los móviles con pantalla grande (sin mencionar a los nuevos *phablets*) permiten algo más que consultas puntuales. Mediante aplicaciones diseñadas a medida, el usuario puede modificar contenidos con cierta comodidad.

Por ejemplo, la aplicación gratuita [Campus](#) permite que los estudiantes de la UOC participen en el campus virtual desde dispositivos móviles con sistema operativo iOS o Android.

Estos dos sistemas operativos son precisamente los que se han impuesto en el mercado y los que deben tenerse en cuenta al desarrollar nuevas aplicaciones.

#### 2.5.1. iOS

iOS es el sistema operativo desarrollado por Apple para los dispositivos móviles iPad e iPhone (además de para otros dispositivos como iPod Touch y Apple TV).

##### Ventajas de iOS

A la hora de desarrollar aplicaciones para dispositivo móvil, el sistema iOS ofrece grandes ventajas.

- Es un sistema robusto. Al ser un sistema operativo comercial, de pago, la compañía Apple está interesada en resolver rápidamente cualquier posible error. Además, como sólo se ejecuta en hardware de la propia compañía los problemas de integración con los dispositivos son mínimos.

- Proporciona una buena experiencia táctil. Aunque otros sistemas han mejorado mucho sus sistemas táctiles, los dispositivos Apple conservan el prestigio de ser los más amigables.
- Dispone de un buen mercado de aplicaciones. Los fabricantes de software saben que los usuarios de productos Apple tienen cultura de pago. Esto es, son usuarios más propensos a pagar por aplicaciones en lugar de descargarlas de manera ilegal. Por ello, iOS suele ser el primer sistema para el que se desarrollan aplicaciones interesantes, algunas gratuitas.

En el ámbito del E-learning, la tienda de Apple incluye un espacio [Apple in Education](#) donde descargar aplicaciones para dispositivo móvil. En especial, la aplicación gratuita [iTunes U](#) permite tanto seguir cursos de los principales MOOC como elaborar contenidos educativos a medida.

### Inconvenientes de iOS

Entre los inconvenientes del sistema iOS, destacan los siguientes:

- Se ejecuta sobre una gama menor de dispositivos. Al estar soportado sólo por productos Apple, no es posible probarlo en otros dispositivos de diferente tamaño o capacidad.
- Es caro. Los dispositivos Apple tienen un precio elevado. Además, la licencia de desarrollador para iOS cuesta 99 dólares al año, mientras la licencia para Android requiere un sólo pago de 25 dólares.

### 2.5.2. Android

El sistema operativo Android desarrollado por Google tiene muchas de las ventajas e inconvenientes del código libre.

#### Ventajas de Android

Entre las ventajas del sistema Android, destacan las siguientes:

- Es gratuito.
- Se ejecuta sobre muchos dispositivos diferentes (porque cualquier fabricante puede utilizarlo como sistema operativo de sus productos).
- Dispone de una gran familia de aplicaciones gratuitas.
- En España, es el sistema operativo utilizado en más dispositivos móviles.
- Dispone de herramientas gratuitas como [J2ObjC](#) que facilitan la portabilidad a iOS de las aplicaciones.
- Puede aprenderse en multitud de tutoriales gratuitos. Dispone incluso de cursos en plataformas MOOC como [Android: Programación de Aplicaciones](#) de Miriada X o [Creative, Serious and Playful Science of Android Apps](#) de Coursera.

Por estas características, Android se presenta como un sistema con menos barreras. Es una plataforma idónea para desarrollar una aplicación de E-learning que los estudiantes puedan ejecutar gratuitamente en multitud de dispositivos.

#### Inconvenientes de Android

Entre los inconvenientes, destaca cierta lentitud en la corrección de *bugs*. Los problemas que surgen en la gran cantidad de dispositivos soportados pueden tardar cierto tiempo en resolverse mediante actualizaciones.

# Capítulo 3

## Diseño

### Contents

---

<b>3.1. Actores</b> . . . . .	<b>18</b>
<b>3.2. Casos de uso</b> . . . . .	<b>18</b>
3.2.1. Alta test . . . . .	18
3.2.2. Ejecución test . . . . .	19
<b>3.3. Interfaces de usuario</b> . . . . .	<b>19</b>
3.3.1. Interfaz alta test . . . . .	19
3.3.2. Interfaz resolución test . . . . .	19
<b>3.4. Interfaces de comunicación</b> . . . . .	<b>20</b>
3.4.1. Interfaz de comunicación web . . . . .	20
3.4.2. Interfaz de persistencia . . . . .	20
<b>3.5. Interfaz software</b> . . . . .	<b>20</b>

---

Para satisfacer los requisitos descritos en la [Introducción](#), se ha diseñado un sistema software con la narrativa y las interfaces gráficas descritas a continuación.

### 3.1. Actores

El sistema diseñado cuenta con dos actores:

- El profesor que elabora las series de test educativos.
- El alumno que cumplimenta los test de cada serie.

### 3.2. Casos de uso

La ejecución del sistema comprende los casos de uso descritos a continuación.

#### 3.2.1. Alta test

Para dar de alta una serie de test, el profesor:

1. Accede a la interfaz de edición de test.

2. Registra los datos de una nueva serie de test (título, descripción, opciones, opción correcta, explicación).
3. Confirma que la serie de test puede ser cumplimentada por los alumnos.

### 3.2.2. Ejecución test

Para cumplimentar una serie de test, el alumno:

1. Arranca la aplicación en su dispositivo móvil.
2. Selecciona la serie de test que desea cumplimentar.
3. En cada test de la serie:
  - a) Selecciona una respuesta.
  - b) Confirma su selección.
  - c) Consulta si ha seleccionado la respuesta correcta.
  - d) Consulta una breve explicación teórica sobre la respuesta correcta.
4. Tras cumplimentar todos los test de una serie:
  - a) Consulta el total de aciertos errores.
  - b) Repite la misma serie de test o selecciona otra.

## 3.3. Interfaces de usuario

El sistema dispone de las interfaces de usuario descritas a continuación.

### 3.3.1. Interfaz alta test

El profesor dispone de un entorno de edición desde donde redactar el contenido de las series de test. En concreto, para cada test de la serie debe añadir los siguientes contenidos:

- Descripción de la pregunta.
- Posibles respuestas.
- Respuesta correcta.
- Breve explicación teórica de la respuesta correcta.

Desde la misma interfaz de edición, el profesor puede confirmar los cambios y dar de alta la serie de test en el sistema para que los alumnos la ejecuten desde sus dispositivos móviles.

### 3.3.2. Interfaz resolución test

En su dispositivo móvil, el alumno dispone de una aplicación gráfica desde donde seleccionar y cumplimentar cada serie de test. Tras responder a cada pregunta, puede verificar si su respuesta es correcta y consultar las explicaciones añadidas por el profesor.

## 3.4. Interfaces de comunicación

El sistema desarrollado dispone de las siguientes interfaces de comunicación.

### 3.4.1. Interfaz de comunicación web

El sistema dispone de una interfaz web para gestionar la transmisión remota de datos:

- Cuando el profesor da de alta un test.
- Cuando el alumno descarga un test en su dispositivo.

### 3.4.2. Interfaz de persistencia

El sistema dispone de una interfaz para gestionar la persistencia en el sistema de los siguientes datos:

- Series de test y explicaciones teóricas.
- Respuestas seleccionadas por el alumno.

## 3.5. Interfaz software

El sistema puede ejecutarse sobre cualquier dispositivo que tenga instalado el sistema operativo Android. No obstante, se podrá especificar una versión mínima soportada.



# Capítulo 4

## Implementación con Android SDK

### Contents

---

<b>4.1. Entorno de desarrollo</b> . . . . .	<b>21</b>
4.1.1. ADT Bundle . . . . .	21
4.1.2. Hoja de cálculo . . . . .	22
<b>4.2. Diagrama de clases</b> . . . . .	<b>22</b>
<b>4.3. Manual de usuario</b> . . . . .	<b>23</b>
4.3.1. Configuración de las series de test . . . . .	23
4.3.2. Ejecución de las series de test . . . . .	24

---

Para implementar el sistema descrito en [Diseño](#) se ha desarrollado una aplicación Android a medida. Este capítulo describe los principales aspectos de dicha aplicación, desde el entorno de desarrollo utilizado hasta el manual de usuario.

### 4.1. Entorno de desarrollo

La aplicación Android se ha programado en lenguaje [Java](#) con las herramientas de desarrollo descritas en las siguientes secciones.

#### 4.1.1. ADT Bundle

Para facilitar la integración del entorno de programación con el de pruebas, el sitio [Android Developers](#) pone a disposición de los programadores el [ADT Bundle](#), un *paquete* de utilidades para el desarrollo.

En concreto, el programa de instalación del ADT Bundle despliega las siguientes herramientas (ya integradas y listas para desarrollar aplicaciones Android).

#### Eclipse

Eclipse es la IDE de programación en Java por excelencia. Además, la versión de Eclipse incluida en el ADT Bundle está especialmente configurada con las librerías del [Android SDK](#) o entorno de desarrollo para Android.

La documentación de dichas librerías puede consultarse en <http://developer.android.com/reference>

### Emulador

El emulador de Android para PC permite probar las diferentes fases de cada proyecto en un dispositivo Android con la versión del sistema y los requisitos de memoria seleccionados.

El emulador incluido en el ADT Bundle puede lanzarse directamente desde Eclipse para evaluar el proyecto en curso.

#### 4.1.2. Hoja de cálculo

El sistema desarrollado en este proyecto final de carrera no implementa el editor que utiliza el profesor para definir las series de test. Para esta tarea se han preferido aprovechar editores de hojas de cálculo y el formato de exportación *XML para Excel 2003*.

En concreto, el formato de exportación *XML para Excel 2003* está soportado por editores como los siguientes:

- El editor [Excel](#) del paquete ofimático [Microsoft Office](#)
- El editor [Libre Office Calc](#) del paquete ofimático gratuito [Libre Office](#).

✓ La sección [Configuración de las series de test](#) detalla los pasos que debe seguir el profesor para definir los test mediante una hoja de cálculo.

## 4.2. Diagrama de clases

Para implementar la aplicación Android se han desarrollado las clases Java del siguiente diagrama UML (figura 4.1).

✓ El apéndice [Package com.uoc.dchiner.TFC](#) incluye una referencia completa de los métodos y atributos de cada clase.

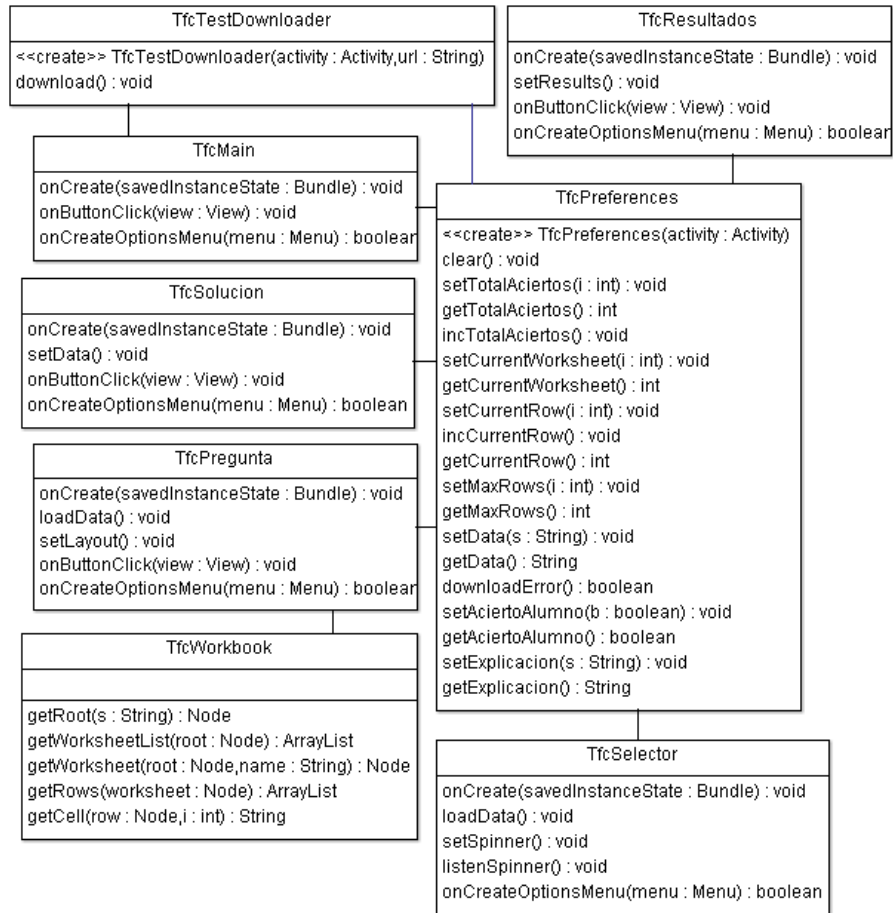


Figura 4.1: Diagrama UML del proyecto.

## 4.3. Manual de usuario

Esta sección detalla los pasos necesarios para configurar una serie de test y ejecutarla desde la aplicación Android.

### 4.3.1. Configuración de las series de test

Las series de test ejecutadas por el alumno desde el dispositivo móvil deben configurarse previamente tal como se explica a continuación.

**Para configurar una serie de test:**

1. Arrancamos un editor de hojas de cálculo como [Microsoft Excel](#) o [Libre Office Calc](#) (figura 4.2).

A	B
Título	Desc
Encaminamiento estático	¿Cuál de las siguientes políticas aplica el encaminamiento?
Metro edge	¿A qué segmento de una red metropolitana corresponde?
Reservas con filtro fijo	En el protocolo RSVP ¿cómo se reservan recursos?

Figura 4.2: Editor de hojas de cálculo Libre Office Calc.

2. Creamos un nuevo documento.
3. A la primera hoja de cálculo del documento, le asignamos el nombre de la primera serie de test (e.g. “Redes de comunicación”).
4. En cada fila de la hoja de cálculo, definimos un test de la serie. Para ello, rellenamos las columnas descritas en el cuadro 4.1.

Columna	Contenido
1	Título del test (e.g. “Encaminamiento estático”)
2	Pregunta del test
3	Primera opción del test
4	Segunda opción del test
5	Tercera opción del test
6	Opción correcta
7	Explicación teórica de la respuesta correcta

Cuadro 4.1: Columnas de cada hoja de cálculo

5. Repetimos la operación para definir las demás series de test en nuevas hojas de cálculo del documento.
6. Exportamos el documento en formato **XML de Microsoft Excel 2003** y codificación UTF-8.
7. Subimos el archivo a un directorio web con acceso HTTP.

#### 4.3.2. Ejecución de las series de test

Desde su dispositivo móvil, el alumno puede seleccionar y ejecutar una serie de test tal como se explica a continuación.

##### Para ejecutar una serie de test:

1. Arrancamos la aplicación desde el dispositivo móvil.
2. En la pantalla principal (figura 4.3), escribimos la URL del archivo que contiene la especificación XML de las series de test.

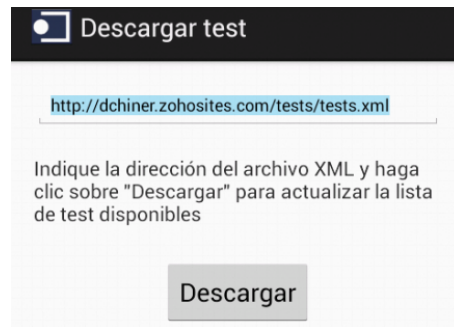


Figura 4.3: Pantalla principal de la aplicación Android.

3. Hacemos clic sobre **Descargar** (figura 4.3) para iniciar la descarga.
4. Si se produce algún error durante la descarga, la aplicación muestra un mensaje de error (figura 4.4) para que volvamos a hacer clic sobre **Descargar**.

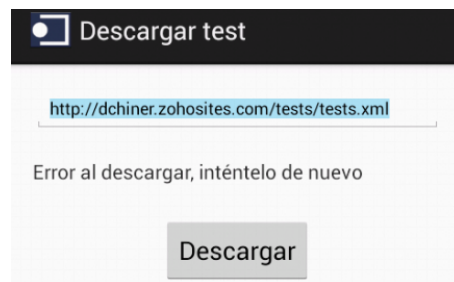


Figura 4.4: Mensaje de error de la aplicación Android.

5. Seleccionamos la serie de test que deseamos ejecutar (figura 4.5).

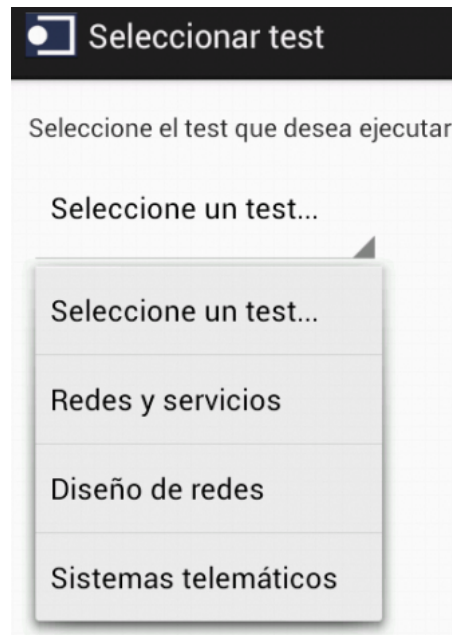


Figura 4.5: Selección de la serie de test en la aplicación Android.

6. En cada test de la serie (figura 4.6), seleccionamos una respuesta y hacemos clic sobre **Siguiente**.

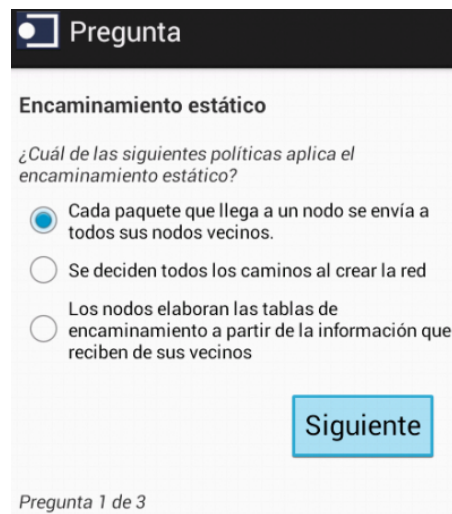


Figura 4.6: Opciones de un test en la aplicación Android.

7. En la pantalla **Solución** (figura 4.7) consultamos nuestro acierto o error y una breve explicación teórica.

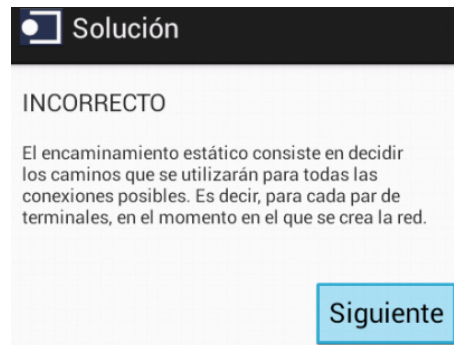


Figura 4.7: Resultado de un test en la aplicación Android.

8. Al final la serie de test (figura 4.8), consultamos la puntuación obtenida y seleccionamos entre las siguientes opciones:
- **Repetir test:** iniciar de nuevo la última serie de test.
  - **Elegir test:** iniciar una nueva serie de test.
  - **Descargar test:** descargar la especificación de otras series de test.

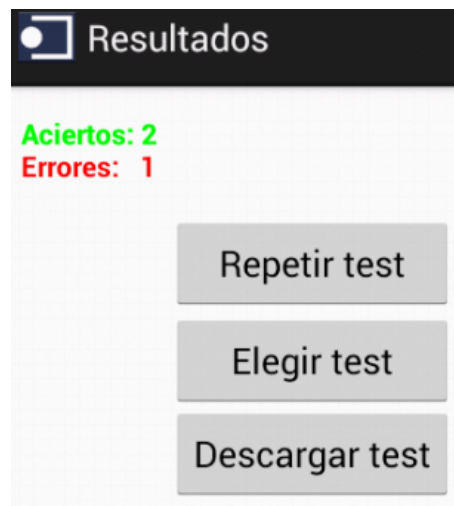


Figura 4.8: Resultados de una serie de test en la aplicación Android.

# Capítulo 5

## Implementación con Android Apps

### Contents

---

<b>5.1. Implementación con doForms</b> . . . . .	<b>28</b>
5.1.1. Elaboración de cuestionarios con doForms . . . . .	29
5.1.2. Ejecución de cuestionarios con doForms . . . . .	30
5.1.3. Consulta de resultados con doForms . . . . .	32
<b>5.2. Implementación con Device Magic</b> . . . . .	<b>33</b>
5.2.1. Elaboración de cuestionarios con Device Magic . . . . .	33
5.2.2. Ejecución de cuestionarios con Device Magic . . . . .	34
5.2.3. Consulta de resultados con Device Magic . . . . .	36
<b>5.3. Comparativa</b> . . . . .	<b>36</b>

---

La tienda online [Google Play](#) permite descargar aplicaciones Android para configurar cuestionarios y ejecutarlos desde dispositivo móvil.

Dichas aplicaciones suelen estar orientadas a entornos profesionales. Por ejemplo, sirven para que un técnico envíe un parte a la sede central tras realizar una instalación a domicilio.

No obstante, las posibilidades de configuración de sus formularios online son lo bastante amplias como para aplicarlos al uso educativo.

Este capítulo evalúa las posibilidades de configuración de dos aplicaciones Android para implementar un sistema como el descrito en [Diseño](#).

✓ Las implementaciones descritas a continuación aprovechan una funcionalidad de las aplicaciones no contemplada en [Diseño](#): enviar los resultados desde los dispositivos para que el profesor los consulte de forma centralizada.

### 5.1. Implementación con doForms

[doForms](#) es una aplicación empresarial para elaborar formularios complejos con campos de texto, herramientas de reporte y varios formatos de preguntas (texto libre, test, multi-selección).

Para implementar un sistema de formularios educativos con [doForms](#), deben seguirse los pasos descritos en las siguientes secciones.



### 5.1.1. Elaboración de cuestionarios con doForms

Esta sección explica cómo configurar cuestionarios mediante doForms.

#### Para elaborar cuestionarios mediante doForms:

1. Accedemos mediante un navegador a [www.mydoforms.com](http://www.mydoforms.com)
2. Creamos una cuenta de usuario:
  - Gratuita, con capacidades limitadas.
  - De pago, con todas las funcionalidades y servicio de soporte.
3. En la pestaña **Build Forms** (figura 5.1) añadimos un nuevo formulario.

The screenshot shows the doForms interface with the 'Build Forms' tab selected. The main content area displays a form titled 'Encaminamiento estático' with a question: '¿Cuál de las siguientes políticas aplica el encaminamiento estático?' and a list of three radio button options. The interface includes a top navigation bar with 'Dispatch', 'View Data', 'Build Forms', 'Projects', 'Mobile Units', and 'Web Users'. Below the navigation bar, there are tabs for 'Redes y servicios', 'Draft', 'File', 'Options', and 'Resources'.

Figura 5.1: Nuevo formulario doForms.

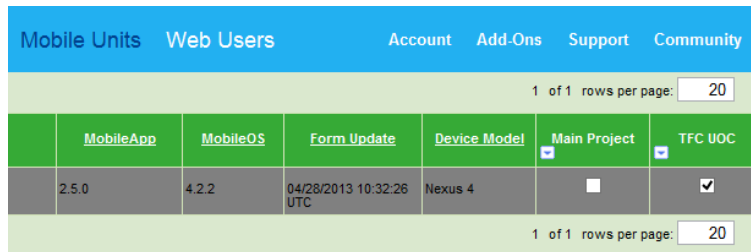
4. En el nuevo formulario, incluimos preguntas tipo test mediante la plantilla *Choose One*.
5. Definimos las condiciones lógicas que establecen cuándo mostrar tanto los mensajes de evaluación (*¡Correcto!*, *Incorrecto*) como las explicaciones teóricas.
6. Definimos los demás formularios.
7. Guardamos los formularios con el estado *Published*.
8. En la pestaña **Projects** (figura 5.2), creamos un nuevo proyecto.

The screenshot shows the doForms 'Projects' interface. It features a table with columns for 'Action', 'Form Name', 'Main Project', and 'TFC UOC'. A context menu is open over the 'Main Project' column, showing options: 'Add project', 'Select all forms', and 'Deselect all forms'. The table lists three projects: 'Photo Survey', 'Timesheet - Daily', and 'Service Request'.

Action	Form Name	Main Project	TFC UOC
<input checked="" type="checkbox"/>	Photo Survey	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Timesheet - Daily	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Service Request	<input type="checkbox"/>	<input type="checkbox"/>

Figura 5.2: Nuevo proyecto doForms.

9. Activamos las casillas correspondientes para añadir nuestros formularios al nuevo proyecto.
10. Confirmamos los cambios.
11. En la pestaña **Mobile Units** (figura 5.3), registramos nuestro dispositivo móvil.



MobileApp	MobileOS	Form Update	Device Model	Main Project	TFC UOC
2.5.0	4.2.2	04/28/2013 10:32:26 UTC	Nexus 4	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 5.3: Nuevo dispositivo móvil en doForms.

12. Activamos la correspondiente casilla para añadir el nuevo proyecto a nuestro dispositivo.
13. Confirmamos los cambios.

### 5.1.2. Ejecución de cuestionarios con doForms

Una vez definidos, los cuestionarios de doForms pueden ejecutarse desde dispositivo móvil tal como se explica a continuación.

**Para ejecutar cuestionarios mediante doForms:**

1. En nuestro dispositivo móvil, instalamos la aplicación [doForms](#) para Android.
2. Arrancamos la aplicación (figura 5.4).

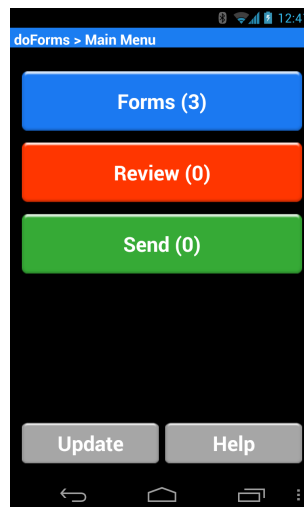


Figura 5.4: Aplicación móvil doForms.

3. Hacemos clic sobre **Update** (figura 5.4) para descargar los últimos test.
4. Hacemos clic sobre **Forms** (figura 5.4) para acceder a los formularios donde hemos definido las series de test (figura 5.5).

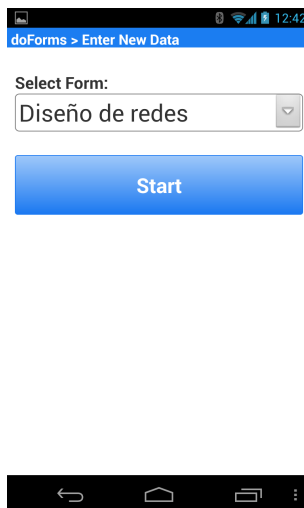


Figura 5.5: Selección de formulario en doForms.

5. En la lista **Select Form** (figura 5.5) seleccionamos el formulario que deseamos ejecutar.
6. Hacemos clic sobre **Start** (figura 5.5) para empezar a responder a los test del formulario seleccionado (figura 5.6).

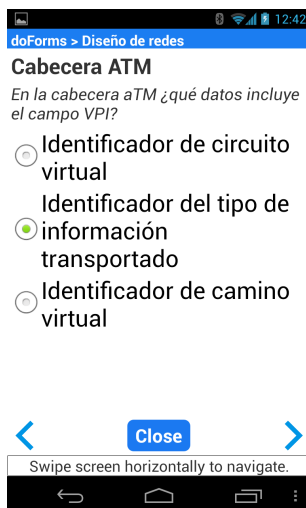


Figura 5.6: Formulario doForms.

7. Tras responder a todos los test, disponemos de diferentes opciones para enviar los resultados o guardarlos en el dispositivo (figura 5.7).

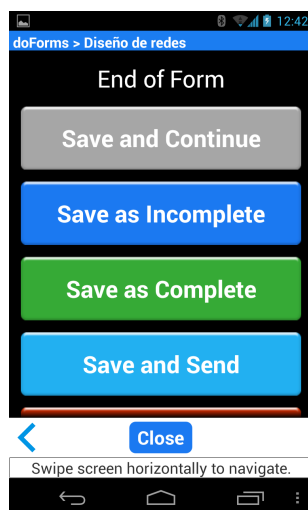


Figura 5.7: Opciones de publicación en doForms.

### 5.1.3. Consulta de resultados con doForms

Las respuestas a los diferentes cuestionarios ejecutados desde dispositivo móvil pueden consultarse de forma centralizado tal como se explica a continuación.

Para consultar los resultados mediante doForms:

1. Iniciamos sesión en [www.mydoforms.com](http://www.mydoforms.com)
2. En la pestaña **View Data** (figura 5.8) seleccionamos un formulario y un rango temporal.

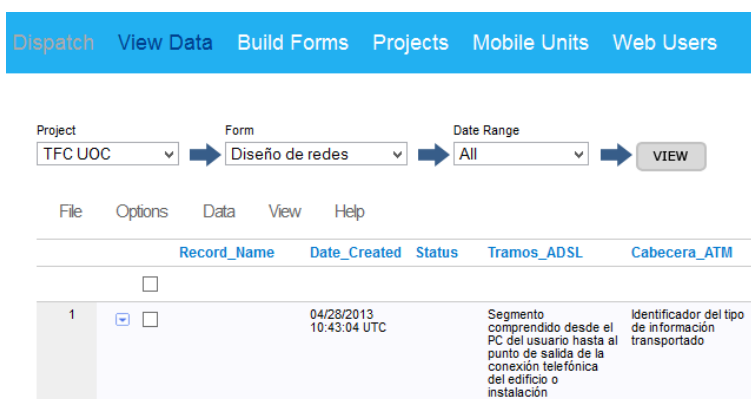


Figura 5.8: Consulta de resultados en doForms.

3. En la lista de resultados, consultamos las respuestas enviadas desde los diferentes dispositivos registrados.
4. Opcionalmente, exportamos los resultados en el formato deseado.

## 5.2. Implementación con Device Magic

Device Magic es una aplicación empresarial para elaborar, rellenar y enviar formularios de seguimiento. Como veremos a continuación, incorpora además una interesante integración con repositorios de publicación externos.

Para implementar un sistema de formularios educativos con Device Magic, deben seguirse los pasos descritos en las siguientes secciones

### 5.2.1. Elaboración de cuestionarios con Device Magic

Esta sección explica cómo configurar cuestionarios mediante Device Magic.

Para elaborar cuestionarios mediante Device Magic:

1. Accedemos mediante un navegador a [www.devicemagic.com](http://www.devicemagic.com)
2. Creamos una cuenta de usuario:
  - Gratuita, con capacidades limitadas.
  - De pago, con todas las funcionalidades.
3. En la pestaña **Forms** (figura 5.9) hacemos clic sobre **New form**.

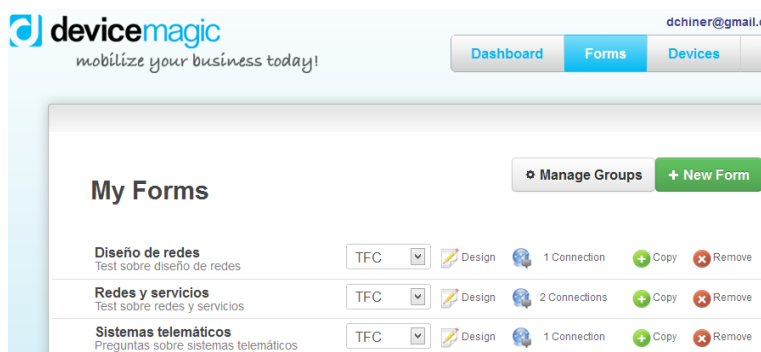


Figura 5.9: Lista de formularios Device Magic.

4. En el editor de formularios (figura 5.10), configuramos las preguntas y respuestas de un test.

✓ Los formularios de Device Magic no soportan secuencias de varios de test y soluciones.

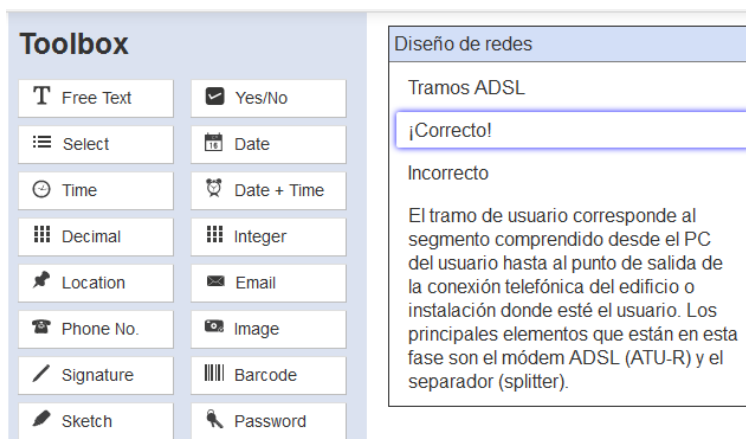


Figura 5.10: Nuevo formulario Device Magic.

- Definimos las condiciones lógicas (figura 5.11) que establecen cuándo mostrar tanto los mensajes de evaluación (*¡Correcto!*, *Incorrecto*) como las explicaciones teóricas.

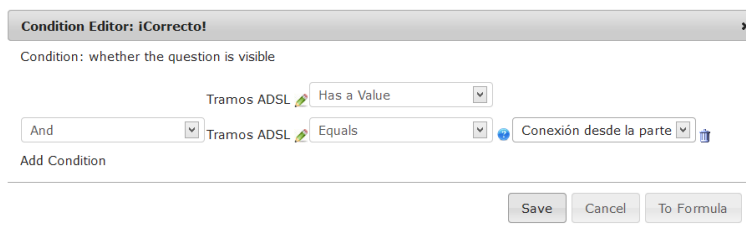


Figura 5.11: Condiciones lógicas de un formulario Device Magic.

- Confirmamos los cambios.
- De vuelta a la pestaña **Forms** hacemos clic sobre **Add Data Connections** para configurar el modo de registrar los resultados.

✓ Device Magic no almacena los datos registrados desde los dispositivos móviles. No obstante, permite registrarlos mediante cuentas externas. Por ejemplo, enviarlos mediante correo o añadirlos a una hoja de cálculo de Google.

- También en la pestaña **Forms**, añadimos el nuevo formulario a un grupo de formularios.
- En la pestaña **Devices**, registramos el dispositivo y seleccionamos el grupo de formularios que podrá ejecutar.

✓ La vista **Devices** también permite consultar un registro de log de las acciones ejecutadas desde el dispositivo.

### 5.2.2. Ejecución de cuestionarios con Device Magic

Una vez definidos, los cuestionarios de Device Magic pueden ejecutarse desde dispositivo móvil tal como se explica a continuación.

Para ejecutar cuestionarios mediante Device Magic:

1. En nuestro dispositivo móvil, instalamos la aplicación **Forms** para Android.
2. Arrancamos la aplicación (figura 5.12).

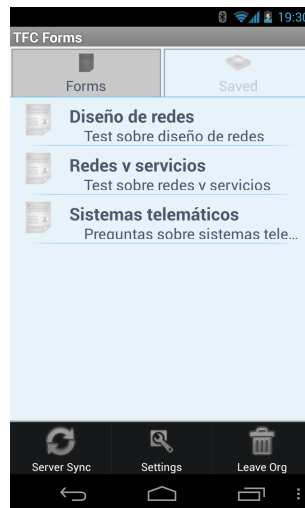


Figura 5.12: Aplicación móvil Forms de Device Magic.

3. Hacemos clic en **Server Sync** (figura 5.12) para actualizar la lista de formularios.
4. Seleccionamos el formulario al que deseamos responder (figura 5.13).

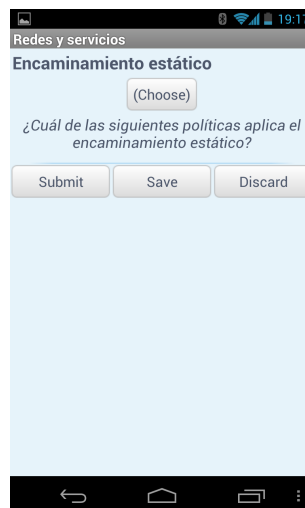


Figura 5.13: Opciones de un formulario de la aplicación Forms.

5. Hacemos clic sobre **Choose** (figura 5.13) para seleccionar la respuesta.

6. Consultamos los resultados (figura 5.14).

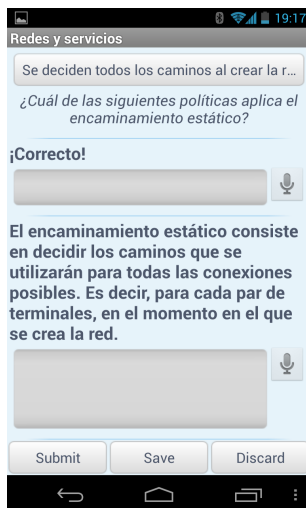


Figura 5.14: Resultados de un formulario de la aplicación Forms.

7. Hacemos clic sobre **Submit** (figura 5.14) para enviar los resultados.

### 5.2.3. Consulta de resultados con Device Magic

Los resultados de los test ejecutados pueden consultarse del modo configurado mediante la opción **Add Data Connections** descrita en [Elaboración de cuestionarios con Device Magic](#).

Por ejemplo, si hemos seleccionado guardarlos en una hoja de cálculo de Google, podemos consultarlos desde [Google Drive](#) (figura 5.15).

submissionid	submissiondatetime	deviceid	username	Tramos-ADSL	-Correcto-	Incorrecto	
							El-tramo-de-usuario-corresponde-al-segmento-comprendido-desde-el-PC-del-usuario-hasta-al-punto-de-salida-de-la-conexión-teléfono-del-edificio-instalación-donde-está-el-usuario- Los-principales-elementos-que-están-en-esta-fase-son-el-módem-ADSL-ATU-R-y-el-separador-splitter-

Figura 5.15: Consulta de resultados en una hoja de cálculo Google.

## 5.3. Comparativa

doForms y Device Magic ofrecen funcionalidades muy parecidas.

Por ejemplo, aunque Device Magic no soporte la consulta de resultados desde su portal web, sí permite consultarlos mediante sistemas externos como Google Drive o servidores de correo.



Device Magic también ofrece más posibilidades de configuración en su versión gratuita, como la configuración de condiciones lógicas.

No obstante, Device Magic sólo permite añadir una pregunta tipo test en cada formulario. De este modo, tras responder a un test y consultar el resultado no es posible pasar al siguiente, es necesario seleccionar un nuevo formulario.

Por ello, para la implementación de este proyecto parecen más adecuadas las funcionalidades que ofrece doForms en su versión de pago.

## Capítulo 6

# Conclusiones y líneas futuras

En el presente trabajo final de carrera se han estudiado las posibilidades de los dispositivos móviles para ejecutar test educativos de autoevaluación. Tanto mediante aplicaciones de gestión de formularios (doForms, Device Magic) como desarrollando una aplicación a medida.

El entorno de desarrollo para Android es lo bastante amigable como para que valga la pena implementar una aplicación a medida y de este modo aprovechar todas las posibilidades de la plataforma. No obstante, el uso de aplicaciones comerciales aporta mucha potencia en el lado servidor, el de la configuración de los test y publicación de los resultados.

Por tanto, a la hora de elegir el sistema definitivo deberán ponderarse factores como los siguientes:

- El precio de contratar los servicios de una aplicación comercial respecto al de desarrollar una aplicación a medida (en los lados clientes y servidor).
- Las posibilidades de configuración de una aplicación comercial respecto a las funcionalidades de una aplicación programada especialmente para el propósito deseado.

El prototipo desarrollado en este trabajo final de carrera aporta una primera visión del *flujo de trabajo* que supone ejecutar un test educativo desde dispositivo móvil. Desde que el profesor define las series de test hasta que el alumno las cumplimenta y evalúa sus resultados.

Para desarrollos futuros, en este flujo de trabajo quedan abiertas las siguientes líneas de mejora:

- En la publicación de las series de test, definir el entorno de edición que utilizará el profesor (como usuario no necesariamente técnico) para elaborar las series de test y confirmar que pueden ser ejecutadas por los alumnos.
- En la ejecución de las series de test, intercalar mejores explicaciones teóricas para que la aplicación sea también una herramienta de consulta.
- En la evaluación de los resultados, mantener la finalidad de autoevaluación (sólo el alumno consulta el número de aciertos o errores) o habilitar también un “modo examen” que publique los resultados en el panel del profesor.

A estas mejoras deberán añadirse obviamente las que propongan los propios usuarios (estudiantes y profesores) al disponer de las primeras versiones de la aplicación.

# Apéndice A

## Package com.uoc.dchiner\_TFC

### Contents

---

<b>A.1. Class TfcMain</b> .....	<b>41</b>
A.1.1. Declaration .....	41
A.1.2. Constructor summary .....	41
A.1.3. Method summary .....	41
A.1.4. Constructors .....	41
A.1.5. Methods .....	41
<b>A.2. Class TfcPreferences</b> .....	<b>42</b>
A.2.1. Declaration .....	42
A.2.2. Constructor summary .....	42
A.2.3. Method summary .....	42
A.2.4. Constructors .....	42
A.2.5. Methods .....	43
<b>A.3. Class TfcPregunta</b> .....	<b>45</b>
A.3.1. Declaration .....	45
A.3.2. Constructor summary .....	45
A.3.3. Method summary .....	46
A.3.4. Constructors .....	46
A.3.5. Methods .....	46
<b>A.4. Class TfcResultados</b> .....	<b>46</b>
A.4.1. Declaration .....	46
A.4.2. Constructor summary .....	46
A.4.3. Method summary .....	47
A.4.4. Constructors .....	47
A.4.5. Methods .....	47
<b>A.5. Class TfcSelector</b> .....	<b>47</b>
A.5.1. Declaration .....	47
A.5.2. Constructor summary .....	47
A.5.3. Method summary .....	48
A.5.4. Constructors .....	48

---

A.5.5. Methods . . . . .	48
<b>A.6. Class TfcSolucion . . . . .</b>	<b>48</b>
A.6.1. Declaration . . . . .	48
A.6.2. Constructor summary . . . . .	48
A.6.3. Method summary . . . . .	48
A.6.4. Constructors . . . . .	49
A.6.5. Methods . . . . .	49
<b>A.7. Class TfcTestDownloader . . . . .</b>	<b>49</b>
A.7.1. Declaration . . . . .	49
A.7.2. Constructor summary . . . . .	49
A.7.3. Method summary . . . . .	49
A.7.4. Constructors . . . . .	50
A.7.5. Methods . . . . .	50
<b>A.8. Class TfcWorkbook . . . . .</b>	<b>50</b>
A.8.1. Declaration . . . . .	50
A.8.2. Constructor summary . . . . .	50
A.8.3. Method summary . . . . .	50
A.8.4. Constructors . . . . .	50
A.8.5. Methods . . . . .	51

---

Este apéndice detalla las clases y métodos del código Java desarrollado para implementar la aplicación Android.

√ La siguiente documentación ha sido generada con <a href="#">TeXDoclet</a> para <a href="#">Javadoc</a> .
--

## A.1. Class TfcMain

Activity principal que permite al usuario indicar una URL desde donde descargar la especificación XML de las series de test.

### A.1.1. Declaration

```
public class TfcMain
extends Activity
```

### A.1.2. Constructor summary

**TfcMain()**

### A.1.3. Method summary

**onButtonClick(View)** Acciones ejecutadas cuando el usuario hace clic en el botón para descargar los test: leer la URL indicada por pantalla, descargar el archivo XML con las series de test, guardar el archivo en las preferencias.

**onCreate(Bundle)** Acciones ejecutadas al crear la clase: inicializar las preferencias.

**onCreateOptionsMenu(Menu)** Acciones ejecutadas para crear el menú (no implementadas).

### A.1.4. Constructors

- **TfcMain**  
public TfcMain()

### A.1.5. Methods

- **onButtonClick**  
public void **onButtonClick**(View view)
  - **Description**  
Acciones ejecutadas cuando el usuario hace clic en el botón para descargar los test: leer la URL indicada por pantalla, descargar el archivo XML con las series de test, guardar el archivo en las preferencias.
  - **Parameters**
    - view – Vista de la interfaz gráfica.
- **onCreate**  
protected void **onCreate**(Bundle savedInstanceState)
  - **Description**  
Acciones ejecutadas al crear la clase: inicializar las preferencias.
- **onCreateOptionsMenu**  
public boolean **onCreateOptionsMenu**(Menu menu)
  - **Description**  
Acciones ejecutadas para crear el menú (no implementadas).

## A.2. Class TfcPreferences

Clase que implementa la persistencia de datos.

### A.2.1. Declaration

```
public class TfcPreferences
extends java.lang.Object
```

### A.2.2. Constructor summary

**TfcPreferences(Activity)** Constructor.

### A.2.3. Method summary

**clear()** Resetea los valores almacenados.

**downloadError()** Indica si se ha producido un error al descargar el documento XML que contiene la definición de los test.

**getAciertoAlumno()** Indica si el alumno ha respondido correctamente a un test.

**getCurrentRow()** Devuelve la posición del test dentro de la serie de test.

**getCurrentWorksheet()** Devuelve la posición de la serie de test ejecutada por el alumno.

**getData()** Devuelve el documento en formato 'XML de Excel 2003' que contiene la definición de los test.

**getExplicacion()** Devuelve la explicación teórica de la respuesta correcta de un test.

**getMaxRows()** Devuelve el número de test incluidos en la serie de test.

**getTotalAciertos()** Devuelve el total de aciertos del alumno respondiendo a los test.

**incCurrentRow()** Incrementa en uno la posición del test dentro de la serie de test.

**incTotalAciertos()** Incrementa en uno el total de aciertos del alumno respondiendo a los test.

**setAciertoAlumno(boolean)** Establece si el alumno ha respondido correctamente a un test.

**setCurrentRow(int)** Establece la posición del test dentro de la serie de test.

**setCurrentWorksheet(int)** Establece la posición de la serie de test ejecutada por el alumno (que en el documento XML corresponde a una hoja de cálculo).

**setData(String)** Guarda el documento en formato 'XML de Excel 2003' que contiene la definición de los test.

**setExplicacion(String)** Establece la explicación teórica de la respuesta correcta de un test.

**setMaxRows(int)** Establece el número de test incluidos en la serie de test.

**setTotalAciertos(int)** Actualiza el total de aciertos del alumno respondiendo a los test

### A.2.4. Constructors

- **TfcPreferences**

```
public TfcPreferences(Activity activity)
```

- **Description**

Constructor.

- **Parameters**

- **activity** – Actividad que invoca las operaciones de la clase

### A.2.5. Methods

- **clear**  
`public void clear()`
  - **Description**  
Resetea los valores almacenados.
- **downloadError**  
`public boolean downloadError()`
  - **Description**  
Indica si se ha producido un error al descargar el documento XML que contiene la definición de los test.
  - **Returns** – 'True' si el documento no ha podido descargarse correctamente, 'False' en caso contrario.
- **getAciertoAlumno**  
`public boolean getAciertoAlumno()`
  - **Description**  
Indica si el alumno ha respondido correctamente a un test.
  - **Returns** – 'True' si el alumno ha respondido correctamente, 'False' en caso contrario.
- **getCurrentRow**  
`public int getCurrentRow()`
  - **Description**  
Devuelve la posición del test dentro de la serie de test.
  - **Returns** – Posición del test dentro de la serie de test.
- **getCurrentWorksheet**  
`public int getCurrentWorksheet()`
  - **Description**  
Devuelve la posición de la serie de test ejecutada por el alumno.
  - **Returns** – Posición de la serie de test ejecutada por el alumno.
- **getData**  
`public java.lang.String getData()`
  - **Description**  
Devuelve el documento en formato 'XML de Excel 2003' que contiene la definición de los test.
  - **Returns** – Documento XML que contiene la definición de los test.
- **getExplicacion**  
`public java.lang.String getExplicacion()`
  - **Description**  
Devuelve la explicación teórica de la respuesta correcta de un test.
  - **Returns** – Explicación teórica de la respuesta correcta de un test.

- **getMaxRows**  
public int getMaxRows()
  - **Description**  
Devuelve el número de test incluidos en la serie de test.
  - **Returns** – Número de test incluidos en la serie de test.
- **getTotalAcertos**  
public int getTotalAcertos()
  - **Description**  
Devuelve el total de acertos del alumno respondiendo a los test.
  - **Returns** – Total de acertos.
- **incCurrentRow**  
public void incCurrentRow()
  - **Description**  
Incrementa en uno la posición del test dentro de la serie de test.
- **incTotalAcertos**  
public void incTotalAcertos()
  - **Description**  
Incrementa en uno el total de acertos del alumno respondiendo a los test.
- **setAcertoAlumno**  
public void setAcertoAlumno(boolean b)
  - **Description**  
Establece si el alumno ha respondido correctamente a un test.
  - **Parameters**
    - b – 'True' si el alumno ha respondido correctamente, 'False' en caso contrario.
- **setCurrentRow**  
public void setCurrentRow(int i)
  - **Description**  
Establece la posición del test dentro de la serie de test.
  - **Parameters**
    - i – Posición del test dentro de la serie de test
- **setCurrentWorksheet**  
public void setCurrentWorksheet(int i)
  - **Description**  
Establece la posición de la serie de test ejecutada por el alumno (que en el documento XML corresponde a una hoja de cálculo).
  - **Parameters**
    - i – Posición de la serie de test ejecutada por el alumno.



**▪ setData**

```
public void setData(java.lang.String s)
```

**• Description**

Guarda el documento en formato 'XML de Excel 2003' que contiene la definición de los test.

**• Parameters**

- s – Documento en formato 'XML de Excel 2003'.

**▪ setExplicacion**

```
public void setExplicacion(java.lang.String s)
```

**• Description**

Establece la explicación teórica de la respuesta correcta de un test.

**• Parameters**

- s – Explicación teórica de la respuesta correcta de un test.

**▪ setMaxRows**

```
public void setMaxRows(int i)
```

**• Description**

Establece el número de test incluidos en la serie de test.

**• Parameters**

- i – Número de test incluidos en la serie de test.

**▪ setTotalAciertos**

```
public void setTotalAciertos(int i)
```

**• Description**

Actualiza el total de aciertos del alumno respondiendo a los test

**• Parameters**

- i – Total de aciertos.

## A.3. Class TfcPregunta

Actividad para mostrar los test a los que debe responder el alumno.

### A.3.1. Declaration

```
public class TfcPregunta  
extends Activity
```

### A.3.2. Constructor summary

```
TfcPregunta()
```

### A.3.3. Method summary

**onButtonClick(View)** Acciones ejecutadas cuando el alumno selecciona una respuesta del test y hace clic en 'Siguiente': guardar la respuesta, evaluar el acierto o error de la respuesta, pasar a la página de solución.

**onCreate(Bundle)** Acciones ejecutadas al crear la clase: cargar las opciones del test, crear la interfaz gráfica.

**onCreateOptionsMenu(Menu)** Acciones ejecutadas al crear el menú (no implementadas).

### A.3.4. Constructors

- **TfcPregunta**  
`public TfcPregunta()`

### A.3.5. Methods

- **onButtonClick**  
`public void onButtonClick(View view)`
  - **Description**  
Acciones ejecutadas cuando el alumno selecciona una respuesta del test y hace clic en 'Siguiente': guardar la respuesta, evaluar el acierto o error de la respuesta, pasar a la página de solución.
  - **Parameters**
    - `view` – Vista de la interfaz gráfica.
- **onCreate**  
`protected void onCreate(Bundle savedInstanceState)`
  - **Description**  
Acciones ejecutadas al crear la clase: cargar las opciones del test, crear la interfaz gráfica.
- **onCreateOptionsMenu**  
`public boolean onCreateOptionsMenu(Menu menu)`
  - **Description**  
Acciones ejecutadas al crear el menú (no implementadas).

## A.4. Class TfcResultados

Actividad para mostrar los resultados finales de una serie de test.

### A.4.1. Declaration

```
public class TfcResultados
extends Activity
```

### A.4.2. Constructor summary

**TfcResultados()**

### A.4.3. Method summary

- onButtonClick(View)** Acciones creadas cuando el alumno hace clic en los diferentes botones: repetir el mismo test, seleccionar un nuevo test, descargar de nuevo la especificación XML de los test.
- onCreate(Bundle)** Acciones ejecutadas al crear la clase: obtener los resultados finales, crear la interfaz gráfica.
- onCreateOptionsMenu(Menu)** Acciones ejecutadas al crear el menú de opciones (no implementadas).

### A.4.4. Constructors

- **TfcResultados**  
`public TfcResultados()`

### A.4.5. Methods

- **onButtonClick**  
`public void onButtonClick(View view)`
  - **Description**  
 Acciones creadas cuando el alumno hace clic en los diferentes botones: repetir el mismo test, seleccionar un nuevo test, descargar de nuevo la especificación XML de los test.
  - **Parameters**
    - `view` – Vista de la interfaz gráfica.
- **onCreate**  
`protected void onCreate(Bundle savedInstanceState)`
  - **Description**  
 Acciones ejecutadas al crear la clase: obtener los resultados finales, crear la interfaz gráfica.
- **onCreateOptionsMenu**  
`public boolean onCreateOptionsMenu(Menu menu)`
  - **Description**  
 Acciones ejecutadas al crear el menú de opciones (no implementadas).

## A.5. Class TfcSelector

Clase para que el alumno seleccione entre los test disponibles.

### A.5.1. Declaration

```
public class TfcSelector
extends Activity
```

### A.5.2. Constructor summary

```
TfcSelector()
```

### A.5.3. Method summary

**onCreate(Bundle)** Acciones ejecutadas al crear la clase: cargar los datos, crear la lista disponible, capturar la selección del alumno.

**onCreateOptionsMenu(Menu)** Acciones ejecutadas al crear el menú (no implementadas)

### A.5.4. Constructors

- **TfcSelector**  
`public TfcSelector()`

### A.5.5. Methods

- **onCreate**  
`protected void onCreate(Bundle savedInstanceState)`
  - **Description**  
Acciones ejecutadas al crear la clase: cargar los datos, crear la lista disponible, capturar la selección del alumno.
- **onCreateOptionsMenu**  
`public boolean onCreateOptionsMenu(Menu menu)`
  - **Description**  
Acciones ejecutadas al crear el menú (no implementadas)

## A.6. Class TfcSolucion

Actividad que muestra los resultados de ejecutar un test.

### A.6.1. Declaration

```
public class TfcSolucion
extends Activity
```

### A.6.2. Constructor summary

**TfcSolucion()**

### A.6.3. Method summary

**onButtonClick(View)** Acciones ejecutadas cuando el alumno hace clic en el botón 'Siguiente': pasar al siguiente test (si no se ha terminado la serie), mostrar los resultados finales (si se ha terminado la serie).

**onCreate(Bundle)** Acciones ejecutadas al crear la clase: construir la interfaz, obtener las respuestas del alumno.

**onCreateOptionsMenu(Menu)** Acciones ejecutadas al crear el menú de opciones (no implementadas).

#### A.6.4. Constructors

- **TfcSolucion**  
public TfcSolucion()

#### A.6.5. Methods

- **onButtonClick**  
public void onButtonClick(View view)
  - **Description**  
Acciones ejecutadas cuando el alumno hace clic en el botón 'Siguiente': pasar al siguiente test (si no se ha terminado la serie), mostrar los resultados finales (si se ha terminado la serie).
  - **Parameters**
    - view – Vista de la interfaz gráfica.
- **onCreate**  
protected void onCreate(Bundle savedInstanceState)
  - **Description**  
Acciones ejecutadas al crear la clase: construir la interfaz, obtener las respuestas del alumno.
- **onCreateOptionsMenu**  
public boolean onCreateOptionsMenu(Menu menu)
  - **Description**  
Acciones ejecutadas al crear el menú de opciones (no implementadas).

### A.7. Class TfcTestDownloader

Clase para descargar mediante conexión HTTP el documento en formato 'XML de Excel 2003' que contiene la especificación de los test.

#### A.7.1. Declaration

```
public class TfcTestDownloader
extends java.lang.Object
```

#### A.7.2. Constructor summary

**TfcTestDownloader(Activity, String)** Constructor

#### A.7.3. Method summary

**download()** Descarga el documento XML.

### A.7.4. Constructors

- **TfcTestDownloader**

```
public TfcTestDownloader(Activity activity, java.lang.String url)
```

- **Description**

Constructor

- **Parameters**

- `activity` – Actividad que inicia la descarga.
- `url` – URL del documento XML

### A.7.5. Methods

- **download**

```
public void download()
```

- **Description**

Descarga el documento XML.

## A.8. Class TfcWorkbook

Clase para gestionar el documento en formato 'XML de Excel 2003' donde se han definido las series de test.

### A.8.1. Declaration

```
public class TfcWorkbook  
extends java.lang.Object
```

### A.8.2. Constructor summary

```
TfcWorkbook()
```

### A.8.3. Method summary

`getCell(Node, int)` Devuelve el valor de una opción de un test.

`getRoot(String)` Devuelve el nodo raíz del documento.

`getRows(Node)` Devuelve la lista de test de una serie de test.

`getWorksheet(Node, String)` Devuelve una serie de test.

`getWorksheetList(Node)` Devuelve la lista de series de test

### A.8.4. Constructors

- **TfcWorkbook**

```
public TfcWorkbook()
```

### A.8.5. Methods

#### ▪ `getCell`

```
public java.lang.String getCell(org.w3c.dom.Node row, int i)
```

- **Description**

Devuelve el valor de una opción de un test.

- **Parameters**

- `row` – Test.
- `i` – Posición de la opción en el test

- **Returns** – Valor de la opción del test.

#### ▪ `getRoot`

```
public org.w3c.dom.Node getRoot(java.lang.String s)
```

- **Description**

Devuelve el nodo raíz del documento.

- **Parameters**

- `s` – Documento.

- **Returns** – Nodo raíz.

#### ▪ `getRows`

```
public java.util.ArrayList getRows(org.w3c.dom.Node worksheet)
```

- **Description**

Devuelve la lista de test de una serie de test.

- **Parameters**

- `worksheet` – Nodo raíz de la serie de test.

- **Returns** – Lista de test de la serie.

#### ▪ `getWorksheet`

```
public org.w3c.dom.Node getWorksheet(org.w3c.dom.Node root, java.lang.String name)
```

- **Description**

Devuelve una serie de test.

- **Parameters**

- `root` – Nodo raíz.
- `name` – Nombre de la serie de test.

- **Returns** – Serie de test.

#### ▪ `getWorksheetList`

```
public java.util.ArrayList getWorksheetList(org.w3c.dom.Node root)
```

- **Description**

Devuelve la lista de series de test

- **Parameters**

- `root` – Nodo raíz

- **Returns** – Lista de series de test.

# Bibliografía

- [1] Josep Playà “Los cursos on line y gratuitos obligan a las universidades a replantear su modelo” [en línea]. La Vanguardia.com. 5 de febrero 2013.  
<<http://www.lavanguardia.com/vida/20130205/54366352711/cursos-on-line-gratuitos-universidades-replantear-modelo.html>>
- [2] Jeremy Hsu “Professor leaving Stanford for online education startup” [en línea]. NBCNEWS.com. 25 de enero de 2012.  
<<http://www.nbcnews.com/id/46138856>>
- [3] M. Mitchell Waldrop “Online learning: Campus 2.0” [en línea]. Nature. 20 de marzo de 2013.  
<<http://www.nature.com/news/online-learning-campus-2-0-1.12590>>
- [4] Asociación para la investigación de los medios de comunicación “Navegantes en la Red - Encuesta AIMC a usuarios de Internet” [en línea]. AIMC. 8 de marzo de 2013.  
<<http://www.aimc.es/-Navegantes-en-la-Red-.html>>
- [5] VVAA “Android vs iOS” [en línea]. Compare Anything. Consultado el 31 de marzo de 2013.  
<[http://www.diffen.com/difference/Android\\_vs\\_iOS](http://www.diffen.com/difference/Android_vs_iOS)>
- [6] StatCounter “Top 8 Mobile Operating Systems in Spain from Oct 2012 to Mar 2013” [en línea]. StatCounter Global Stats.  
<[http://gs.statcounter.com/#mobile\\_os-ES-monthly-201210-201303](http://gs.statcounter.com/#mobile_os-ES-monthly-201210-201303)>



# Índice alfabético

ADT Bundle, [21](#)  
Android, [6–8](#), [16](#), [17](#), [20–22](#), [28](#), [40](#)  
Android SDK, [21](#)  
Apple, [16](#), [17](#)

Chris Dede, [13](#)  
clear(), [43](#)  
COMA, [13](#)  
Coursera, [13–15](#), [17](#)  
CREDIT, [14](#)

Device Magic, [33](#), [34](#), [36](#)  
DiMarzio, [8](#)  
doForms, [28–30](#), [36](#)  
download(), [50](#)  
downloadError(), [43](#)

E-learning, [9](#), [11](#), [12](#), [14](#), [16](#), [17](#)  
Eclipse, [21](#), [22](#)  
Emulador, [22](#)  
Excel, [22–24](#)

Facebook, [15](#)  
FAO, [9](#)  
Francesco Cirillo, [10](#)

getAciertoAlumno(), [43](#)  
getCell(Node, int), [51](#)  
getCurrentRow(), [43](#)  
getCurrentWorksheet(), [43](#)  
getData(), [43](#)  
getExplicacion(), [43](#)  
getMaxRows(), [44](#)  
getRoot(String), [51](#)  
getRows(Node), [51](#)  
getTotalAciertos(), [44](#)  
getWorksheet(Node, String), [51](#)  
getWorksheetList(Node), [51](#)

Google, [6](#), [9](#), [10](#), [17](#)  
Google Drive, [15](#), [36](#)  
Google Forms, [15](#)  
Google Play, [6](#), [28](#)  
Google+, [15](#)

Harvard, [13](#)

Imma Turbella, [13](#)  
incCurrentRow(), [44](#)  
incTotalAciertos(), [44](#)  
IOC, [12](#)  
iOS, [16](#), [17](#)  
iPad, [16](#)  
iPhone, [16](#)

Java, [21](#), [22](#), [40](#)  
Javadoc, [40](#)  
Jesús Conde, [8](#)  
Jordi Torres, [12](#)

LaTeX, [10](#), [16](#)  
LateX Labs, [16](#)  
Libre Office, [22](#)  
Libre Office Calc, [23](#)  
LMS, [14](#)

Microsoft Office, [22](#)  
Miríada X, [13–15](#), [17](#)  
MOOC, [13–15](#), [17](#)  
Moodle, [14](#)

Nature, [13](#)

onButtonClick(View), [41](#), [46](#), [47](#), [49](#)  
onCreate(Bundle), [41](#), [46–49](#)  
onCreateOptionsMenu(Menu), [41](#), [46–49](#)

Piazza, [15](#)

Pomodoro, [10](#)

Sakai, [14](#)

Sebastian Thrun, [13](#)

setAciertoAlumno(boolean), [44](#)

setCurrentRow(int), [44](#)

setCurrentWorksheet(int), [44](#)

setData(String), [45](#)

setExplicacion(String), [45](#)

setMaxRows(int), [45](#)

setTotalAciertos(int), [45](#)

Stack Overflow, [9](#)

Stanford, [13](#), [14](#)

TfcMain, [41](#)

TfcMain(), [41](#)

TfcPreferences, [42](#)

TfcPreferences(Activity), [42](#)

TfcPregunta, [45](#)

TfcPregunta(), [46](#)

TfcResultados, [46](#)

TfcResultados(), [47](#)

TfcSelector, [47](#)

TfcSelector(), [48](#)

TfcSolucion, [48](#)

TfcSolucion(), [49](#)

TfcTestDownloader, [49](#)

TfcTestDownloader(Activity, String), [50](#)

TfcWorkbook, [50](#)

TfcWorkbook(), [50](#)

Tobias Oetiker, [9](#)

UAB, [12](#)

UdA, [12](#)

Udacity, [13](#)

UML, [22](#)

UOC, [12–14](#), [16](#)

UPC, [12](#)

Verbosus, [16](#)

XML, [22](#), [24](#)

YouTube, [8](#), [15](#)

