

# Marvin Agüero - UGr - DSS - 2016-17

<b>Práctica 5: Aplicación</b>	<b>2</b>
Enunciado	2
Funcionales	2
No funcionales	3
Recursos utilizados	3
Arquitectura y diseño	3
Modelo Vista Controlador (MVC)	3
Patrón escuchador de eventos	4
El juego	4
Temática	4
Preguntas	4
Modo de juego	5
Instalación	5
Generación de .apk	5
Tests	5
Fuentes	6
Recursos	6
Anexos	6
Pantallas	6

# Práctica 5: Aplicación

## Enunciado

Desarrollo completo de una aplicación interactiva para dispositivos móviles: patrón escuchador de eventos

- El juego que se tiene que desarrollar ha de poder ejecutarse en un dispositivo Android (versión  $\geq 4.4$ ) o IOS (elegir sólo uno) y se trata de presentar una serie de preguntas al usuario, que tendrá que responder para alcanzar una puntuación global, así como el número de respuestas acertadas y falladas.
- Se tendrá una primera pantalla de presentación con un botón para iniciar el juego, otro para obtener los resultados y estadísticas obtenidos después de jugar y un tercer botón para enlazar con juegos similares libres que existan en Internet y que se puedan utilizar desde un tipo de aplicación como la que se pretende desarrollar.
- Cada pantalla del juego ha de consistir en 1 pregunta con 4 posibles respuestas alternativas; la pregunta podría ser un texto plano, o una pregunta que contenga imágenes y sonidos.
- Se ha de construir una base de datos de preguntas de texto planas a la que acceda directamente la app, de tal forma que en el futuro se puedan cambiar las preguntas de texto sin tener que modificar el código de la citada aplicación, aunque haya que volver a generar el archivo apk desplegable.
- Al seleccionar una de las respuestas en cada pregunta ha de aparecer un mensaje de felicitación en el pie de la pantalla junto con un sonido de acierto o fallo; después de unos segundos se pasará a la siguiente pantalla con otra pregunta.
- En caso de no acertar, se ha de poder elegir entre volver a la pantalla inicial para comenzar de nuevo el juego y se obtendrá un mensaje con la respuesta correcta o bien continuar el juego (no se obtendría las soluciones hasta el final); en ambos casos, se pueden obtener los resultados acumulados durante el desarrollo de un juego volviendo a la pantalla de inicio y pulsando el botón "Resultados".
- Tras la última pregunta, el mensaje que se mostrará será de felicitación por haber realizado el juego y mensaje de despedida antes de volver a iniciar el juego.

Para que la aplicación desarrollada sea evaluada favorablemente ha de satisfacer los siguientes requisitos:

## Funcionales

1. El juego deberá mostrar una batería de preguntas.
2. El usuario ha de poder responder a las preguntas de una en una
3. Si el usuario falla la pregunta, se le mostrará un mensaje indicándole que ha fallado junto con un sonido
4. Si el usuario acierta, se le muestra un mensaje indicándole que ha acertado.
5. Ha de haber pantallas de pregunta que incluyan 1 imagen

6. Si el usuario falla, entonces deberá aparecer la opción de continuar la partida o de volver a iniciar el juego.
7. si el usuario acierta, continuará con la siguiente pregunta.
8. Cuando el usuario finaliza la partida, se le muestran los resultados que ha obtenido.
9. El menú deberá poder dirigirnos, al inicio de una nueva partida, a una opción que nos permita ver los resultados obtenidos en las partidas realizadas anteriormente, o a otros juegos en la Red mediante un webView.

## No funcionales

1. Solamente puede hacerse un único acceso a la base de datos.
2. La aplicación deberá funcionar como mínimo en dispositivos con versiones de sistemas operativos actuales (por ejemplo: Android  $\geq 4.4$ )
3. La aplicación debe funcionar tanto en dispositivos móviles (teléfonos) como en tablets.
4. Se debe desarrollar la aplicación pedida aplicando Patrones de Diseño Software (por ejemplo: Singleton, Inmutable, Proxy, Escuchador de Eventos) y Patrones Arquitectónicos (por ejemplo: MVC, DataBaseManagement System).

## Recursos utilizados

- Windows 10 64 bits. 8 GB RAM, Intel i7 3ª Generación, 2400 Ghz
- Android Studio 2.2.3
- [Intel HAXM](#)

## Arquitectura y diseño

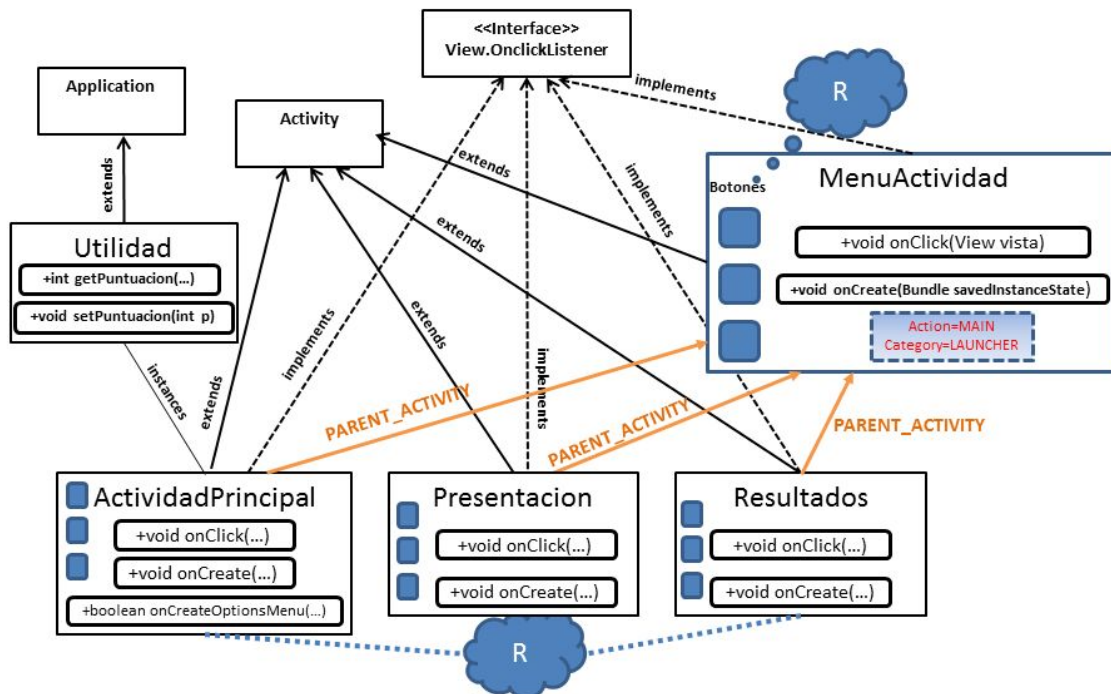
### Modelo Vista Controlador (MVC)

La aplicación sigue una arquitectura clásica MVC:

- Modelo:
  - Bases de datos,
  - imágenes y
  - sonidos
- Vista
  - Layouts,
  - Styles
  - Strings
- Controlador
  - Middleware con bases de datos: clase DBHelper.java
  - Middleware con las vistas: Actividades de Android
  - Modelo interno seguido: Preguntas

## Patrón escuchador de eventos

Como se ve en la Figura 1, toda la aplicación gira en torno alrededor de la clase *ActividadPrincipal*, y el patrón se aplica sobre las actividades de la aplicación que implementan la interfaz *View.OnClickListener*, como la clase *Pregunta*, *Utilidad*, *Despedida* o *MenuActividad*, llamadas según la necesidad.



Fuente: Desarrollo de una aplicación para dispositivos móviles con Android, M.I. Capel, DSBCS, MII, Ugr, 2016.

## El juego

### Temática

Se trata de un juego Quiz, denominado **QuizApp** con preguntas sobre películas animadas con un público orientativo hacia los niños pero no excluyente, para smartphones Android desde 4.4 Kitkat (API level 20).

### Preguntas

Son 30 preguntas de tres tipos:

- Pregunta de texto con opciones de respuesta tipo texto: 9
- Pregunta de texto con opciones de respuesta tipo imagen: 3
- Pregunta sobre una imagen con opciones de respuesta tipo texto: 15

- Pregunta sobre un sonido con opciones de respuesta tipo texto: 3

## Modo de juego

La aplicación abarca todos los requisitos funcionales y no funcionales solicitados para la práctica, a parte de los ya mencionados incluyen lo siguiente:

- Conjunto de preguntas aleatorias, 10 en total
- Recuperadas de una base de datos en texto plano
- 4 posibles respuestas
- Se responden una por una
- Tras un fallo, suena un sonido de error, se puede continuar (se restan dos puntos) o empezar de vuelta el quiz
- Tras una cierto, suena un sonido de éxito con mensaje.
- Al finalizar las preguntas del conjunto, suena un sonido de finalización, se muestra puntuación conseguida
- En todo momento se puede empezar una nueva tanda, consultar puntuación de ronda anterior o abrir otros juegos pero en navegador
- Único acceso a la base de datos (al crearse actividad principal).

## Instalación

Se puede instalar usando el archivo app-release.apk, aunque se debe habilitar la instalación de aplicaciones de orígenes desconocidos en el smartphone objetivo, generalmente al copiar al teléfono o descargarlo y quererlo ejecutar, pregunta si deseamos habilitarlo.

## Generación de .apk

Android nos permite generar el signed .apk de una forma amigable, entonces seguí estas [instrucciones](#) para ello. Este .apk generado será compatible con versiones iguales o posteriores al API 20 (Android 4.4 Kitkat).

## Tests

En simulador de Android + Intel HAXM, indicando utilizar GPU, que al configurar el dispositivo, en opciones avanzadas se indica, utilizar hardware:

- Smartphones: Nexus\_6\_API\_25, 5 Pulgadas, 560dpi, 1GB RAM, Android 7.1
- Tablet: Nexus\_6\_API\_25, 9 Pulgadas, xhdpi, 1GB RAM, Android 7.1

Además se instaló el .apk release y se probó en un dispositivo real verificando su correcto funcionamiento: Samsung Grand Prime 4G, 1GB RAM, 5 Pulgadas, Android 5.1

## Fuentes

Android development with Android Studio - Tutorial, Lars Vogel, 2009, vogella GmbH, version 14.4, 20.06.2016. Recuperado de <http://www.vogella.com/tutorials/Android/article.html>

CinemaQuiz, @fblupi, 2016. Recuperado de <https://github.com/fblupi/CinemaQuiz>

Desarrollo de una aplicación para dispositivos móviles con Android, M.I. Capel, DSBCS, MII, Ugr, 2016.

Desarrollo completo de una aplicación interactiva para dispositivos móviles, M.I. Capel, DSBCS, MII, Ugr, 2016.

GoogleSamples, Google, 2016. Recuperado de <https://github.com/googlesamples?utf8=%E2%9C%93&q=quiz&type=&language=>

Seriator, @germaan, 2016, <https://github.com/germaan/Seriator>

## Recursos

los archivos multimedia que utiliza la aplicación son todos de licencia libre y extraídos de:

<https://openclipart.org/>  
<https://www.freesound.org/>  
<https://pixabay.com>  
<https://archive.org>  
<http://www.upsocl.com/>

## Anexos

### Pantallas

