Trivialix

Práctica 2 – Laboratorio de dispositivos móviles

Autores:

* Marcos Sánchez Hernández – 981 – GIS
* Sara Rodríguez Alarcón – 781 – GII
* Lucía Colás Vico – 1007 – GIS

ÍNDICE

[INTRODUCCIÓN 2](#_Toc57024608)

[DESCRIPCIÓN DE LA APLICACIÓN 3](#_Toc57024609)

[BASE DE DATOS 4](#_Toc57024610)

[PANTALLAS DE LA APP Y LA RELACIÓN ENTRE ELLAS 5](#_Toc57024611)

[PRINCIPALES CLASES UTILIZADAS 8](#_Toc57024612)

[DIFICULTADES ENCONTRADAS Y SU SOLUCIÓN 8](#_Toc57024613)

[CONCLUSIONES 8](#_Toc57024614)

[BIBLIOGRAFÍA 8](#_Toc57024615)

# INTRODUCCIÓN

En esta práctica, perteneciente al bloque número dos de la asignatura Laboratorio de dispositivos móviles denominado: Gestión de elementos multimedia y gestión de base de datos, se ha pedido realizar una aplicación en Android Studio que tuviera efectos de sonido, reproducción de sonidos y persistencia de datos.

A continuación, se expondrá una breve descripción de la aplicación realizada, de los elementos utilizados para su correcto funcionamiento y de la persistencia de datos.

# DESCRIPCIÓN DE LA APLICACIÓN

La aplicación desarrollada ha recibido el nombre de Trivialix y consiste en un juego de preguntas donde los usuarios pueden elegir la temática que quieran y realizar un cuestionario de 10 preguntas para comprobar su conocimiento en cualquiera de las seis temáticas que hay para escoger: cine, música, series, deportes, cine y deportes electrónicos (e-Sports).

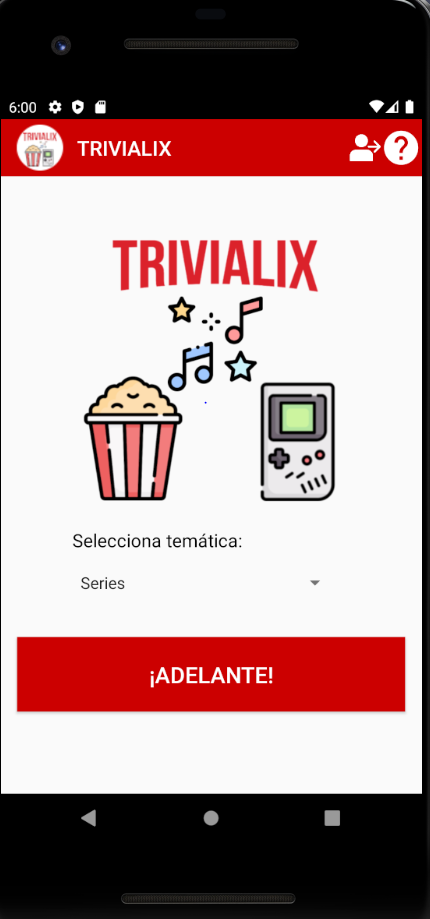
Los usuarios también tienen la opción de registrarse en la aplicación para ver si se encuentran entre los que más saben y que su puntuación quede guardada para intentar superarla en la misma temática o en cualquier otra temática.

Ilustración 1: Pantalla de inicio de Trivialix

# BASE DE DATOS

Al ser un juego de preguntas se ha optado por utilizar una base de datos SQL aprovechando que Android Studio tiene soporte para SQLite el cual está basado en dicho lenguaje.

La base de datos ha sido creada fuera de la aplicación con el entorno SQLiteBrowser y añadida posteriormente a la aplicación para evitar tiempos de carga mayores a la hora de probar la aplicación debido a la gran cantidad de preguntas que contiene pero también permite la creación de nuevos usuarios una vez que la aplicación ya está iniciada y guardándolos en la base de datos.

Las tres entidades que han sido utilizadas en la base de datos han sido las siguientes: temáticas, preguntas y usuarios. La relación entre usada entre preguntas y temáticas es de 1..n debido a que una pregunta solo puede pertenecer a una temática pero una temática puede tener varias preguntas. La entidad usuario es independiente a las demás dado que solo guarda el nombre de usuario, su contraseña y su puntuación.

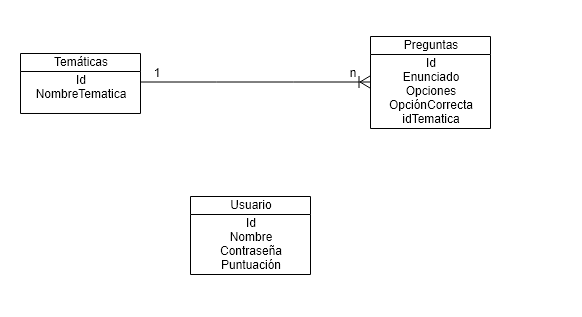


Ilustración 2: Modelo Entidad-Relación

# PANTALLAS DE LA APP Y LA RELACIÓN ENTRE ELLAS

A continuación, se mostrarán las pantallas de las que se compone la aplicación y mediante un diagrama de navegación se expondrán sus relaciones.

* Inicio

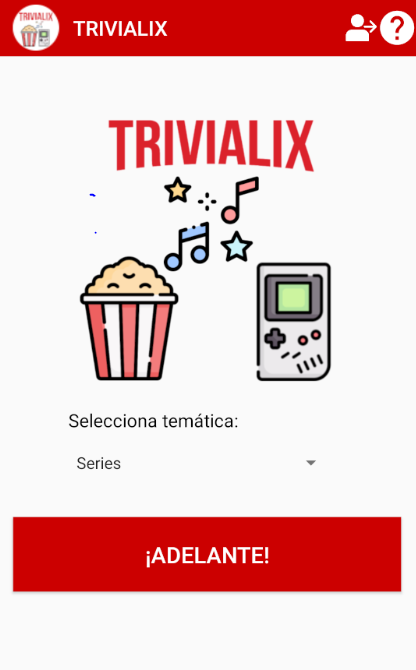


Ilustración 3: Pantalla de inicio

* Ayuda

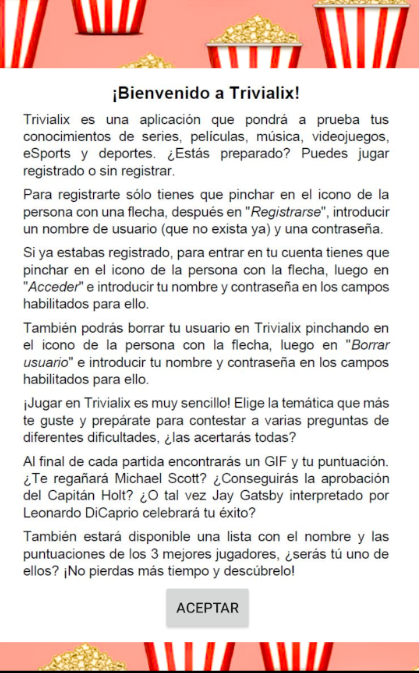


Ilustración 4: Pantalla de ayuda

* Iniciar sesión/Registro/Borrar usuario

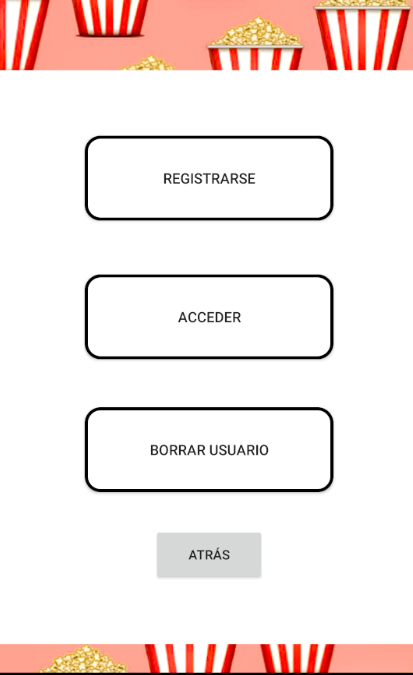


Ilustración 5: Pantalla de inicio de sesión/Registro/Borrar Usuario

* Preguntas

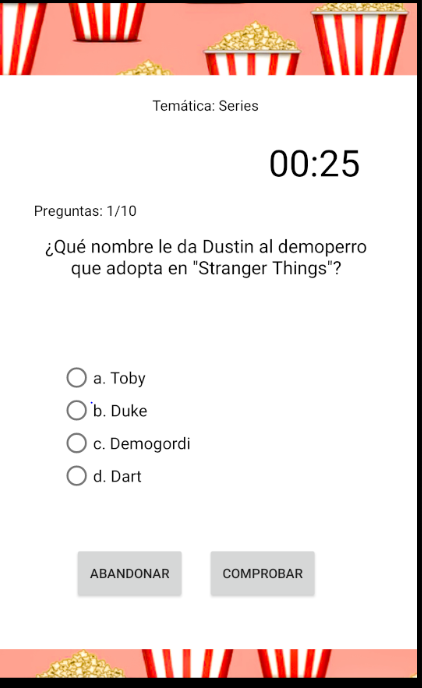


Ilustración 6: Pantalla de preguntas

* Mostrar puntuación

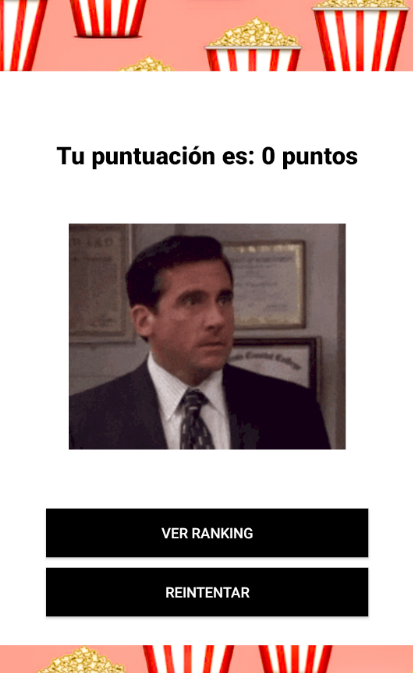


Ilustración 7: Pantalla de puntuación

* Ranking

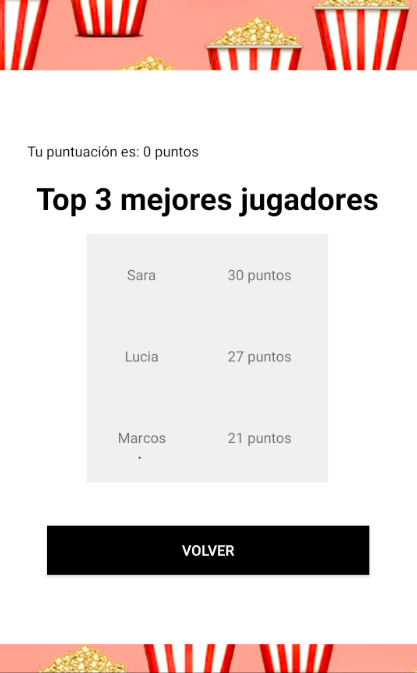


Ilustración 8: Pantalla de ranking

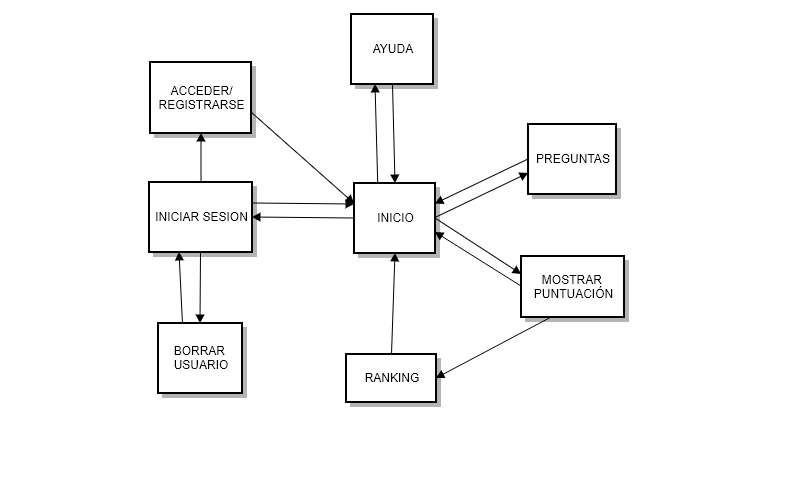


Ilustración 9: Diagrama de navegación

# PRINCIPALES CLASES UTILIZADAS

# DIFICULTADES ENCONTRADAS Y SU SOLUCIÓN

* Una de las dificultades encontradas fue la de sustituir el *ActionBar* por el *Toolbar* y añadir dentro de este todos los elementos deseados (el nombre de la aplicación, el logo, el botón de registro y el botón que nos lleva al manual). Para resolverlo, primero añadimos en el *styles.xml* “*Theme.AppCompat.Light.NoActionBar*”, de esta manera ocultamos el *ActionBar*, seguidamente creamos un *xml* denominado “*toolbar*” y otro llamado “*menu*” en donde indicamos todos los elementos que ha de contener el *toolbar*. Finalmente en la *MainActivity* añadimos el logo de la aplicación haciendo uso de *“.setLogo()”,* método que nos permite posicionar a la izquierda del *toolbar* el logo deseado.

# CONCLUSIONES

# BIBLIOGRAFÍA