

PROGRAMACIÓN MULTIMEDIA Y DISP. MÓVILES

App Final de Android



Desarrollado por Cegoco11

Índice

<i>Introducción</i>	<i>2</i>
<i>Comentarios sobre las soluciones técnicas</i>	<i>3</i>
<i>Posibles ampliaciones futuras.....</i>	<i>5</i>
<i>Manual de usuario.....</i>	<i>6</i>

Introducción

La aplicación está basada en *Covone's Revenge*, ya que no dio tiempo a acabarla tal y como se había planeado.

Se puede decir que *Phone Quest* (así es como se llama la app), está dirigida a todos los públicos, pero más concretamente a personas que están aprendiendo a hacer aplicaciones en *Android*. Esto se debe a que en la aplicación se utilizan varios *broadcasts receiver*. A continuación, voy a explicar a grandes rasgos, en que consiste la app.

En la aplicación hay una serie de misiones, que se basan en los *broadcasts receiver*, en las que, interactuando con el móvil, se completan y mandan una notificación avisando de que un logro se ha completado.

¿Por qué digo que el principal público son los alumnos?

Porque el código fuente está, en gran mayoría, comentado y gracias a esto podrían acabar añadiendo más tipos de misiones utilizando diferentes *broadcasts* además de entender fácilmente que hace la app. Y porque en una aplicación se han juntado el uso de *broadcasts*, el uso de *notificaciones*, *almacenamiento interno*, *shared preferences*, el uso de *herencia* y la creación de un *layout* por código, dependiendo de los objetos que se tengan en una lista y teniendo en cuenta sus características.

Comentarios sobre las soluciones técnicas

Lo primero que hice fue dibujar un boceto sobre como quería que fuese la aplicación.

Después de eso, pase a buscar imágenes y música por internet, teniendo en cuenta el *theme* que había decidido, el Antiguo Egipto (Gracias a Oliver).

Cuando ya tenía el material necesario me puse a recortar las imágenes y sonidos con herramientas de edición. He tenido que hacer muchos ajustes con los niveles de sonido y la duración de los mismos a lo largo de todo el desarrollo de la *app*, al igual que efectos de desenfoque en las imágenes para que no quedase demasiado llamativo el fondo de algún *background*.

Una vez hecha la primera pantalla que veía el usuario me puse a hacer la clase Logro (que ha tenido varias modificaciones a lo largo del desarrollo), y a crear el layout según una lista de estos Logros (el cual está generado desde código). También empecé a implementar los diferentes *broadcasts* (tipos de misiones).

El siguiente paso fue, juntar la creación de misiones asociando a cada una un *broadcast* y creando la función *click* dependiente de cada tipo de logro y su *id*.

Los logros se fueron añadiendo en el siguiente orden:

- 1) Conectar los auriculares
- 2) Enchufar el teléfono a la corriente (Facilitado por Jesús)
- 3) Cargar el móvil X porcentaje (se puede configurar la X)
- 4) Descargar el móvil Y porcentaje (se puede configurar la Y)
- 5) Activar el *bluetooth* (Facilitado por David)

A partir de aquí, que la aplicación iba cogiendo forma, me planteé el esquema de guardado. ¿Cuándo se deben guardar los logros en un fichero?

Pues en la primera interacción con la *app* se tienen que crear los logros por defecto que se va a encontrar el usuario nada más entrar a la *app*. Con lo cual he tenido que implementar un método de guardado y un método de carga.

Respondiendo a la pregunta, cuando se cierra la aplicación se tiene que guardar y cuando se completa uno también se debe de guardar.

En esta parte del desarrollo los logros tenían 2 estados, pero no eran visibles hacia el usuario (dejando de lado el hecho de que al completar un logro llegaba una notificación al móvil del usuario). Al final acabé planteando que los logros tenían que tener 3 estados, (activo, nuevo y completado), y que cada uno tenía que tener un impacto visual en la *app*. Además de eso, implemente un método para ordenar el *array* según el estado y según el tiempo de creación de los logros (Esto del tiempo de creación se explicará más adelante).

Ahora que ya tenía el sistema de guardado, me faltaba el sistema de reactivación de logros. Esto lo que viene a ser, es que cuando el usuario cierra la *app*, los *broadcasts* se desactivan. Y si tenemos algún logro activado al reiniciar la *app* se va a ver solo visualmente, pero el *broadcast* seguiría desactivado, así que hice

una función en la que se recorrían los logros, y si el estado era de activado, se volvía a activar ese *broadcast*.

Ahora que la *app* ya tenía casi la forma final, pensé en hacer un menú con tres botones, para que tuviese un poco más de sentido todo, y pensando en futuras actualizaciones. Tuve que volver a buscar imágenes, crear un estilo para los botones y encontrar una canción que pegase para el menú principal.

En este punto de la *app*, me puse a reflexionar sobre la interacción entre el usuario y el móvil, y como conseguir que fuese más *user-friendly*. Aquí fue cuando me planteé usar *AlertDialog* personalizados, así que me puse a investigar sobre cómo hacer una plantilla, poner un fondo personalizado etc...

Al final tuve que acabar usando un *Dialog* a secas, porque en el otro no dejaba personalizarlo tal y como quería.

Ahora ya tenía diálogos de confirmación al aceptar las misiones, y diálogos de error de los que hablaré ahora.

En cuantos, a los tipos de misiones, había dos de ellas que podían no completarse en ciertos casos. Hablo de las misiones de cargar y descargar, que si la misión es de cargar un 5% y tenemos más de un 95% no se debería poder aceptar esa misión. Cuando suceden estos casos, sale un mensaje de error en un diálogo personalizado, avisando al usuario de que esta acción no se puede llevar a cabo.

Retomando el tema del tiempo de creación de los logros, en la etapa final de la *app*, he logrado hacer que se generen misiones aleatorias (dentro de las que existen obviamente), y la fecha de creación sería para que, en el apartado de Estadísticas, se pueda ordenar por fecha de creación, este punto se profundiza un poco más en el siguiente apartado del documento.

Para hacer que el sistema de misiones aleatorias funcione, aunque se cierre la aplicación completamente, he utilizado el *shared preferences*. Cuando pasa el tiempo (actualmente un minuto para hacer el *testing*), aparece un mensaje cuando se pasa por el menú principal (*Dialog*), avisando de que se ha recibido una nueva misión.

El siguiente problema con el que me encontré fue que, al tener varias misiones del mismo tipo, tal y como tengo planteada la *app*, he tenido que añadir la condición de que solo pueda estar activa a la vez, una del mismo tipo.

Lo que me dedique a hacer en esta última fase fue revisar código, corregir algunos *warnings* que decía el *Android studio*, optimizar código, usando herencia y funciones, y hacer *testing* para encontrar posibles bugs.

Posibles ampliaciones futuras

En cuanto a las futuras ampliaciones de la *app*, creo que he conseguido parametrizarla de tal forma que, si se quiere añadir otro tipo de logro, solo habría que cambiar unas pocas líneas. Con lo cual se podrían añadir la mayoría de broadcast teniendo en cuenta si hay que darles permiso, y si realmente son viables/tienen sentido en esta aplicación.

En las notificaciones de la *app* me hubiese gustado hacer un *summary* para poder agruparlas y no crear un posible spam si se cumplen varias misiones seguidas.

En cuanto a los *warnings* que da *Android studio*, me gustaría haber corregido la mayoría de ellos.

Me hubiese gustado hacer un apartado de créditos hacia los autores que han hecho el arte o la música de la aplicación. De todas formas, tengo un fichero de texto en el que les doy crédito, el cual está dentro del proyecto de la aplicación.

También hubiese añadido un apartado de estadísticas en el que se contabilizaran los logros completados, agrupados por tipo y el total de los mismos, además de añadir un historial de los últimos logros, mostrando la fecha en la que han sido completados.

Los idiomas de la *app* es otro tema en el que me hubiese gustado meterme, haciendo un botón en el que se le preguntase al usuario si quiere cambiar de idioma, y mostrarle una lista con las traducciones que hay de la *app*.

Manual de usuario

Lo primero que se va a encontrar el usuario nada más abrir la aplicación es una pantalla con el título de la aplicación (*Imagen1*) y al pulsar sobre esta le llevará al menú principal (*Imagen2*)



Imagen1

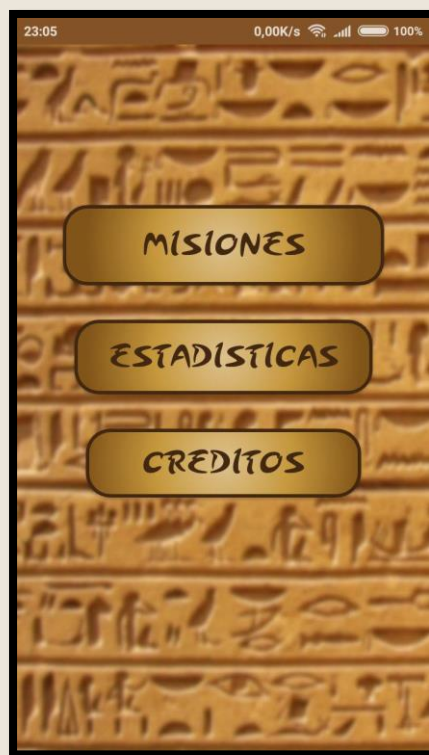


Imagen2

En el menú principal, hay 3 botones:

- Misiones: que nos llevará a la base de nuestra aplicación.
- Estadísticas: que no será funcional hasta futuras actualizaciones.
- Créditos: que no tampoco será funcional hasta futuras actualizaciones.

Al intentar acceder a los que están incompletos, aparecerá un mensaje avisando que están estos apartados están en construcción. (*Imagen3*)

En el menú principal, también puede aparecer un mensaje avisando de que se ha encontrado una nueva misión. Actualmente está configurado para que cada 60 segundos se cree una nueva misión, aunque el usuario cierre completamente la app. (*Imagen4*)



Imagen3



Imagen4

Una vez pulsamos el botón Misiones nos encontramos con la siguiente interfaz (Imagen5), en la que al hacer *tap* encima de las misiones no sale un dialogo de confirmación. (Imagen6)



Imagen5

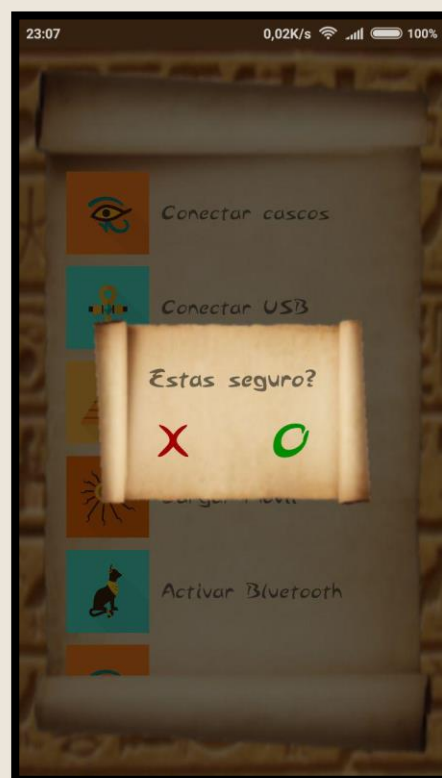


Imagen6

Al pulsar en el botón verde, activaríamos la misión elegida, y el usuario vería el estado "misión activa" como se ve en la siguiente ilustración (Imagen7).

Una vez que se complete la misión, en este caso la de conectar el USB, nos llegaría una notificación avisándonos, además de cambiar visualmente, como se aprecia en las siguientes imágenes. (Imagen8, Imagen9)



Imagen7

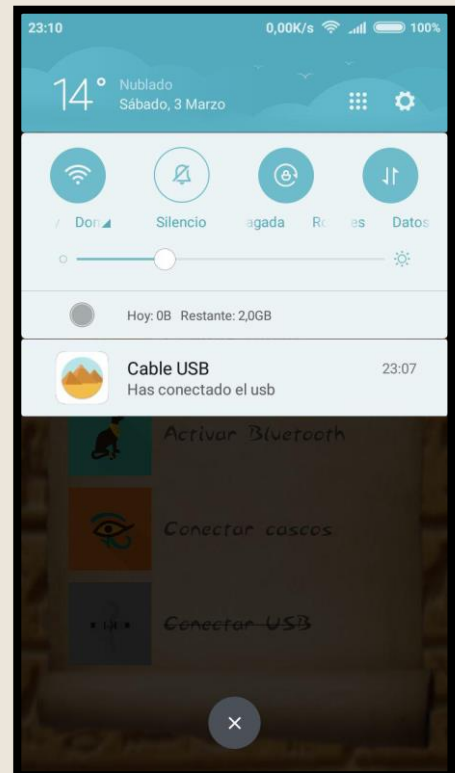


Imagen8

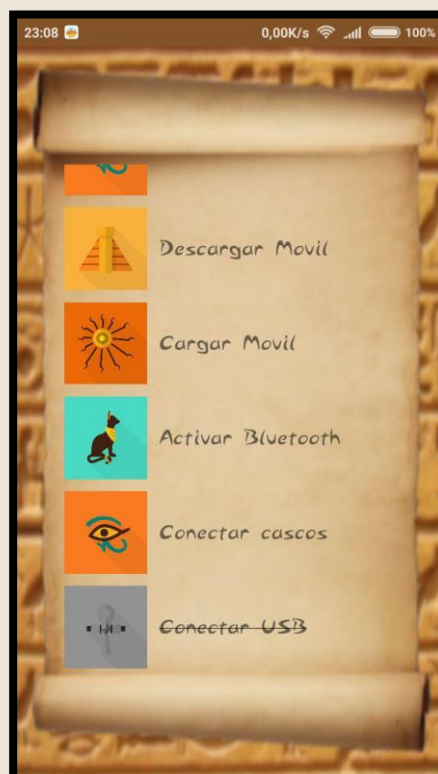


Imagen9