

cashCONTROL

Aplicación para el control de gastos

resumen

El objetivo de cashCONTROL es poder capturar de forma fácil un gasto para su posterior consulta. El programa no pretende ser un sistema de gestión contable. El principal propósito de la aplicación es hacer que la captura (o generación) de un gasto sea sencilla y se pueda realizar con un par de clics. La posibilidad de asociar información de audio a un gasto hace que sea muy fácil pasar gastos. Posteriormente, esos gastos pueden ser consultados para su inclusión en la contabilidad de cualquier organización o particular.

1. Introducción

1.1 Justificación del TFC

Han sido varios los motivos que me han llevado a decantarme por Android como proyecto de fin de carrera. Uno de ellos, quizás el de más peso, es la gran demanda que hay en el mercado. La posibilidad de ofrecer desarrollo en Android a algunos de mis actuales clientes ha tenido un peso importante.

Además del interés en trabajar en Android para terceros, también estaba motivado en realizar una aplicación de control de gastos para uso personal. Esa aplicación que yo necesitaba ya está finalizada y ahora sólo me falta mejorarla para poder comercializarla.

1.2 Experiencia Previa

Antes de realizar el proyecto no disponía de ninguna experiencia de entorno Android, ni siquiera a nivel de usuario, ya que adquirí mi teléfono Android justo cuando empezaba el semestre. La API básica de Java sí que la conocía por haber trabajado en proyectos de J2EE. Conocía algunos patrones de diseño que he podido implementar con éxito.

1.3 Objetivos

El principal objetivo del proyecto realizar una aplicación sencilla pero sólida. Me idea ha sido la de intentar trabajar de forma vertical explorando todas las capas del framework de Android y codificar de la forma más "ortodoxa" posible. Para ello he elegido un proyecto relativamente sencillo. En el futuro me interesará depurar mi aplicación, ampliarla y si tiene aceptación entre un grupo de usuarios, publicarla en el Android Market.

1.4 Planificación

La planificación se ha llevado a cabo siguiendo el método de diseño basado en el usuario.

Las diferentes fases del proyecto han sido:

PAC1 Plan de trabajo - entrega el 11/3/2013.

PAC2 Análisis funcional i diseño técnico – entrega el 8/4/2013.

PAC3 Implementación. – entrega el 20/5/2013.

Entrega final – 10/6/2013.

1.5 Recursos y herramientas utilizadas.

Para este proyecto se han utilizado los siguientes recursos:

- Hardware:

- Ordenador MacBook Pro con sistema operativo Mac OS X 10.7.5
- Teléfono Samsung Galaxy S GT-I9000 con versión de Android 4.1.2

- Entorno de desarrollo integrado:

- Eclipse bundle ADT

- Software de soporte:

- MS Word para la redacción del proyecto.
- Balsamiq Mockups para el diseño de los componentes gráficos.
- Cacao: Herramienta online para creación de diagramas UML

- Otros recursos:

- Página para creación de colores <http://www.2createawebsite.com>
- Repositorio subversion para mantenimiento del código en el sitio gratuito de Assembla: <http://www.assembla.com/home>

1.6 Producto obtenido.

Una vez realizado el proyecto se han producido los siguientes productos:

- Éste mismo documento a modo de síntesis de todo el trabajo realizado.
- Repositorio con los ficheros del código fuente de Java y archivo ejecutable (extensión APK).
- Video de presentación de la aplicación.

La aplicación obtenida es un sistema para la captura de gastos. El programa permite al usuario asociar una nota de audio a cada gasto y guardar de forma automática la fecha y la hora.

Requisitos del dispositivo en el que se va a utilizar la aplicación:

- Sistema operativo Android 2.2 o superior.
- Tarjeta SD para almacenamiento de bases de datos y archivos de audio.

Permisos que requiere del usuario:

- Grabación a través del micrófono.
- Almacenamiento en tarjeta SD

2. Memoria del TFC.

2.1 Diseño.

2.1.1 Diseño de la Base de Datos

La base de datos consta de una única tabla: Gastos

campo	tipo
_id	INTEGER (primary key)
texto	TEXT
tipo	TEXT
importe	NUMERIC
timestamp	INTEGER
longitud	TEXT
latitud	TEXT

No ha sido necesario utilizar herramientas de soporte para SQLite ya que la tabla es sencilla, no hay relaciones y los datos se han podido leer e insertar desde la aplicación con mucha facilidad.

2.1.2 Diseño de la interface gráfica.

La pantalla principal de la aplicación es la que se muestra a continuación. Su diseño está inspirado en una calculadora y un teléfono. La interface está preparada para que el usuario no pueda introducir datos por error (por ejemplo, el botón verde "enviar gasto" sólo se habilita cuando los datos son correctos y están listos para su envío). El importe máximo es de 9999.99 euros y es leído a través de sharedPreferences. El botón backspace (el de la flecha hacia la izquierda) permite editar el texto. El comportamiento del botón clear ("C") está basado en el de una calculadora y su misión es borrar el importe introducido así como cualquier información de audio que se haya podido asociar. Los botones de la parte superior se utilizan para la grabación del audio.



El botón "lista" permite ir a la lista de gastos en cualquier momento. Cualquier importe que se haya editado o audio asociado permanecerán en la activity principal cuando regresemos de la lista. La pantalla principal no dispone de OptionsMenu ya que la aplicación no dispone de parámetros de configuración. La activity que aparece al pulsar "lista" es la siguiente:



36 5:43

▶ Listado de gastos

viernes 21/06/2013

02:07	47.0 €
02:05	28.36 €

jueves 20/06/2013

10:24	12.6 €
10:24	4.0 €
10:23	104.25 €

miércoles 19/06/2013

12:47	5.16 €
12:45	53.41 €

Los gastos aparecen listados agrupados por fechas (en la parte superior las fechas más recientes) y ordenados por horas. Sobre la lista se puede realizar scroll vertical. Al hacer clic sobre un gasto se abre la ficha de gasto que se muestra a continuación:



▶ Ficha de gasto

47.0 €

viernes 21/06/2013
02:07

Escuchar Audio

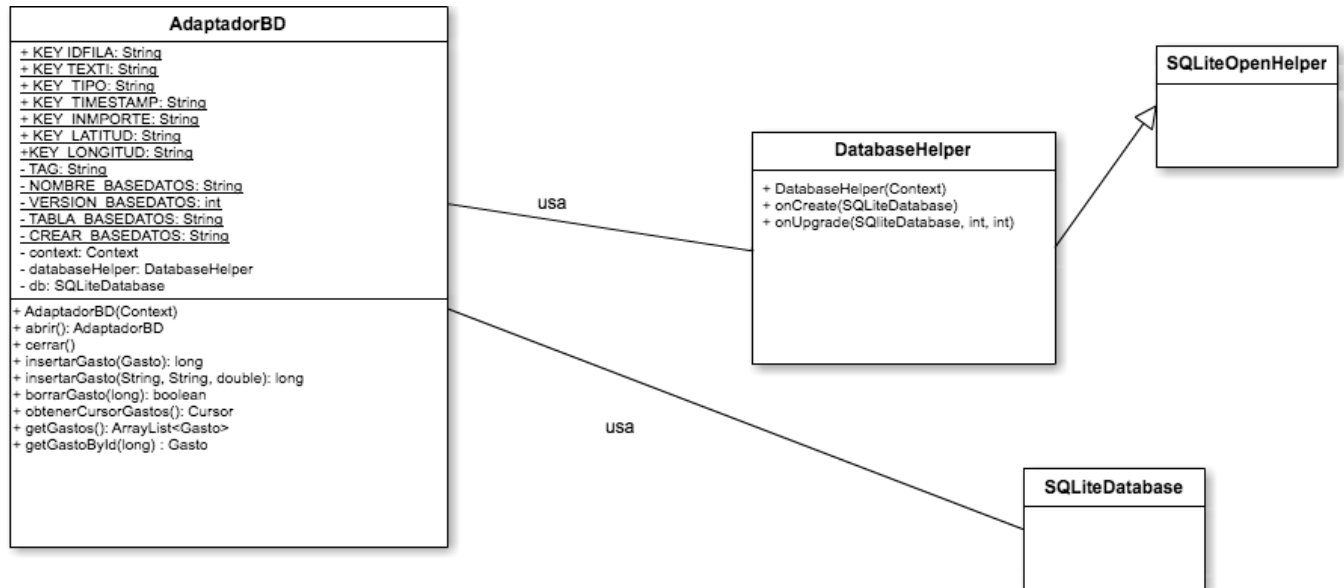
Esta ficha muestra la información básica (importe, fecha, hora y audio en caso de haberlo). Sólo cuando el gasto tiene asociado un audio se muestra el botón "Escuchar Audio" que permite llevar a cabo la audición. El menú OptionsMenu permite eliminar el gasto directamente en la base de datos.

2.2 Implementación

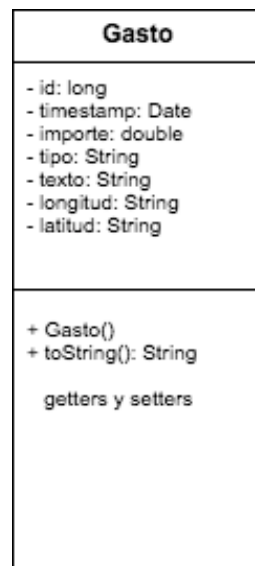
He distribuido las clases en los siguientes packages:

- com.tfc.cashcontrol (para las activities y custom views)
- com.tfc.cashcontrol.basedatos (para la clase helper de acceso a datos)
- com.tfc.cashcontrol.modelo (para los POJOs o "entity classes")

2.2.1 Clase AdaptadorBD.



2.2.2 Clase Gasto.



Esta clase es la única que corresponde al modelo de datos. Dispone de getters y setters para acceder a los atributos. También hay un toString() que resulta cómodo

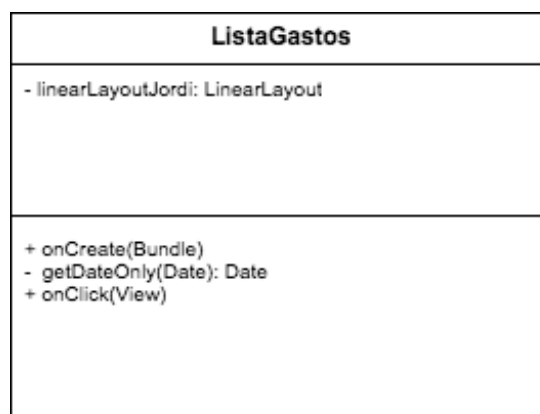
para tareas de Log. No he considerado necesario sobreescribir los métodos equals()/hashCode() ni hacer que la clase implemente el interface Serializable.

2.2.3 Clase FilaGasto.



Esta clase es utilizada por la Activity ListaGastos para mostrar cada uno de los gastos de la base de datos. FilaGasto es un LinearLayout que contiene la información básica de un gasto (hora e importe).

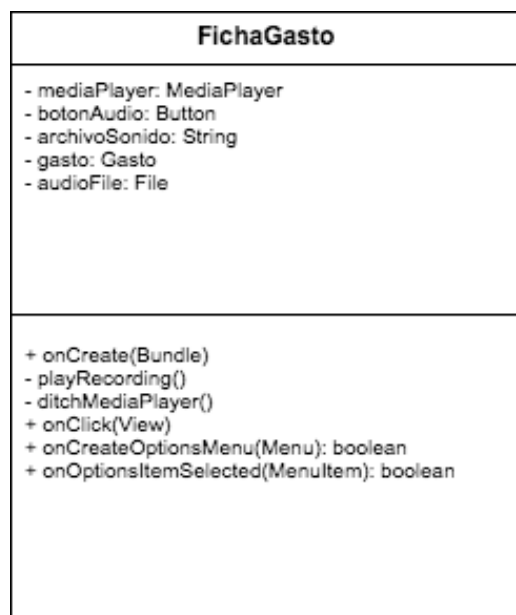
2.2.4 Clase ListaGastos.



Esta clase es una Activity. Su misión es mostrar los gastos en forma de lista sobre la que se puede realizar scroll vertical. Los gastos aparecen agrupados por fechas en

orden descendente y cada grupo de fechas viene ordenado por horas (de forma que el gasto más reciente es el que aparece arriba). El método privado getDateOnly(Date) se encarga de eliminar de una fecha la información de hora (igualando todas las fechas a 00:00) para poder comparar fechas y montar los grupos.

2.2.5 Clase FichaGasto.



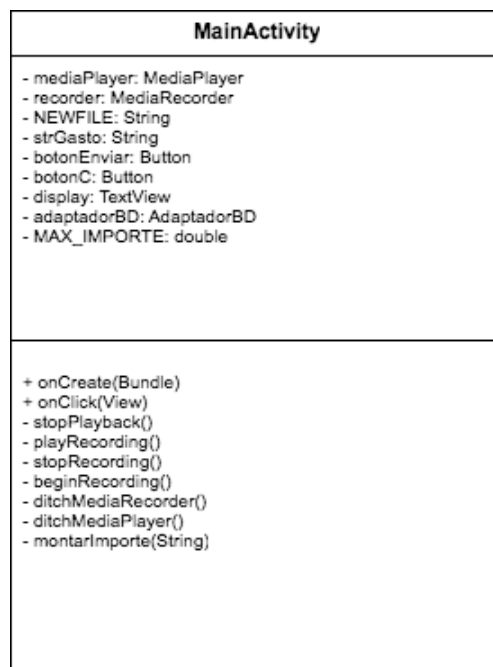
Esta clase es una Activity que tiene como objetivo mostrar la ficha correspondiente a un gasto. Cuando el gasto lleva asociada una información de audio se muestra el botón "Reproducir Audio" que utiliza el método `playRecording()` para escuchar la nota de voz. Los métodos que reproducen audio también están codificados en la clase `MainActivity` y son los mismos (no hay clase helper para la gestión del audio).

A través de Options Menu se puede llevar a cabo la acción de eliminar un gasto.

2.2.6 Clase MainActivity.

Esta es la clase principal de la aplicación. Los métodos privados `stopPlayback()`, `playRecording()`, `stopRecording()`, `beginRecording()` y `ditchMediaPlayer()` se utilizan

para la grabación/reproducción de audio. El método montarImporte(String) recoge las cifras pulsadas desde los botones del teclado de la calculadora para ir montando el importe que se visualizará en el display (y que se acabará guardando en la base de datos como gasto). Esta clase crea una instancia de adaptadorDB para guardar los datos en SQLite. Los archivos de audio se guardan en una carpeta en la tarjeta SD y están asociados con los registros de la base de datos a través del nombre de archivo.



2.3 Evaluación.

He dispuesto de poco tiempo y no he podido realizar pruebas en condiciones para poder evaluar el producto. De todas formas, mi idea es seguir con el proyecto, añadiendo las prestaciones pendientes de implementación así como mejoras en el código.

3. Variaciones con respecto a la idea inicial.

Las principales variaciones que ha sufrido el proyecto respecto a la idea original vienen dadas por la falta de tiempo (por causas imprevistas que no me han permitido cumplir con la planificación prevista).

- Inclusión de tipos de gasto: La idea original era poder asociar a un gasto un tipo ya predefinido.

- Inclusión de texto: Otra prestación que no se ha implementado es la de posibilitar la inclusión de textos editados por el usuario asociados a cada gasto. No considero que sea una función imprescindible para esta aplicación, ya que la inclusión de texto no es rápida y está fuera de la filosofía de cashCONTROL.

- Inclusión de imagen: Esta es la función originalmente definida y no incluida que tiene mayor repercusión en la calidad final del producto. Así como el audio es rápido de incluir y aporta mucha información, la posibilidad de incluir imágenes habría sido ideal (una imagen vale más que mil palabras).

- Imposibilidad de de eliminar varios gastos a la vez. La aplicación en la actualidad sólo permite eliminar gastos de uno a uno.

- No hay preferencias. Algunos aspectos de la aplicación podrían estar configurados como preferencias (por ejemplo, el importe máximo de un gasto, el tiempo máximo de grabación de audio o el símbolo de moneda).

4. Conclusiones.

TFC ha sido la asignatura que he esperado con ganas durante toda la carrera. La oportunidad era única para poder introducirme en Android de una forma acelerada. Durante el inicio del semestre he participado de forma activa en el aula pero a las pocas semanas casi he tenido que abandonar el proyecto por un problema de enfermedad familiar que me ha robado todo el tiempo destinado al estudio. Mi aplicación ha sufrido "recortes" en algunas de sus funciones más flamantes (realización de fotos, geolocalización de los lugares donde se realiza el gasto, etc...) y la calidad de mi trabajo ha bajado en picado durante estas últimas semanas.

Pero al margen de lo expuesto anteriormente, también he sacado conclusiones positivas. Una de ellas, aunque parezca contradictorio, es que mirando mi código actual observo una serie de malas prácticas que me invitan a aplicar mejoras. Veo soluciones a mi código que se podrían haber realizado de forma más eficiente y elegante. Pero creo que el hecho de darme cuenta de estas deficiencias implica que empiezo a conocer el framework de Android.

Otra conclusión positiva es que estoy en vías de adquirir los conocimientos necesarios para desarrollar aplicaciones para Android y de hecho ya estoy en contacto con algunos potenciales clientes.

5. Líneas futuras de trabajo.

Dado el "recorte" de prestaciones que ha sufrido mi aplicación, la principal línea futura consistirá en implementar todo aquello que estaba reflejado en el documento original. Una vez completada esta fase, se podrán aplicar las siguientes mejoras:

- Posibilidad de que los usuarios almacenen los datos en un servidor y los puedan consultar a través de Internet.
- Sistema de mapas para la localización de los puntos en los que se realizan gastos.
- Mejora del diseño de la interface gráfica.
- Configurar settings para el idioma, el tipo de moneda, el tiempo máximo de grabación de audio o el importe máximo del gasto.
- Posibilidad de que la propia app genere un archivo de Excel
- Inclusión de información estadística en gráficos.

6. Bibliografia

- Wikipèdia. *Android* <http://es.wikipedia.org/wiki/Android>
- Pàgina para desarrolladores de Android: <http://developer.android.com>
- Head First Android Development de editorial O'reilly
- Android in Action de editorial Manning
- Programming Android de editorial O'reilly
- Android apps with Eclipse de editorial Apress