UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación en Android para la gestión personal de numismática de euros

Projecte Final de Carrera

Ingenieria Tècnica d'Informàtica de Gestió

Autor: Marcos Fernández Andreu

Director: José Vicente Busquets Mataix

Septiembre 2013

Resumen

Desde que apareciera el primer teléfono móvil estos han sufrido una rapidísima evolución, llegando a convertirse en la actualidad en pequeños computadores de gran potencia que ofrecen un mundo de posibilidades. En la actualidad los llamados *smartphone* ofrecen multitud de servicios como gestores de correo o navegación, cosa que hace que la gente pase gran tiempo usándolos. Además, su tamaño, y el hecho de ser el propio teléfono, hacen que la gente los lleve siempre encima. Por tanto, los *smartphone* se han convertido en una nueva oportunidad de negocio y una herramienta multifuncional muy potente. Es por ello que la proliferación de aplicaciones que realizan funciones concretas se ha disparado en estos años.

En el caso concreto de este trabajo el objetivo es el coleccionismo de monedas de la Zona Euro. Los distintos países de la Zona Euro tienen la capacidad de acuñar monedas con diseño propio, cosa que ha llevado a una extensa gama de monedas diferentes que son objetivo de coleccionistas. Tal es así que se acuñan monedas de 2€ con motivos conmemorativos tales como aniversarios o acontecimientos especiales. Con la aplicación desarrollada se crea un eficiente gestor para coleccionistas, un gestor con funcionalidad concreta que permite identificar las monedas y gestionar la colección personal. Cabe señalar también que si bien existen aplicaciones similares, estas son de pago.

A continuación se explica más detalladamente el proceso de desarrollo de la aplicación, los distintos componentes de la misma así como de la tecnología empelada. Por último se señalaran algunas mejoras y se extraerán algunas conclusiones.

Palabras clave

Android, aplicación móvil, numismática, gestión

Tabla de contenidos

Tabla de imágenes

1. Introducción

1.1 Motivación

El creciente *boom* de los *Smartphone* y tablets, así como del sistema operativo de Google, Android, ha abierto un nicho de mercado muy atractivo. Además, el hecho de ser un concepto novedoso hace que nuevas generaciones puedan introducirse desde un primer momento. Por otro lado, la tecnología sobre la que se basa Android es Java, un lenguaje multiplataforma, libre y cada vez más extendido. Dado que la filosofía de Android es también una filosofía abierta, existen multitud de comunidades de desarrolladores independientes que facilitan ayuda. Por todo ello, he considerado Android como una atractiva opción para desarrollar esta aplicación para la colección de monedas de 2€

1.2 Objetivos y alcance

El objetivo principal de este proyecto es desarrollar una sencilla, pero eficiente, aplicación para gestionar una colección personal de monedas de la zona euro. Se pretende tener un control sobre las monedas que se tienen en posesión, poder visualizarlas y saber cuáles faltan en nuestra colección.

Para ello se ha optado por Android en su forma más básica, es decir, sin librerías de terceros. En un primer instante se ha estudiado Android desde cero, familiarizándose con sus componentes más básicos y su forma de funcionar. Hechas algunas pruebas básicas, y habiéndose familiarizado también con el IDE Eclipse, se ha comenzado el análisis y el posterior desarrollo de la propia aplicación.

Para el análisis se han descrito unas sencillos requisitos que se entienden son básicos para este proyecto. También se han realizado unos sencillos diagramas para los casos de uso. Esto ha ayudado a comprender el objetivo de la aplicación así como la forma que finalmente debía tener.

Posteriormente se ha procedido al desarrollo puro, a escribir el código fuente. Se han puesto en práctica todas las nociones básicas de programación vistas y todos los nuevos conocimientos adquiridos sobre Android. Se ha intentado tener un mínimo de variedad en sus elementos para poder comprobar como de potente puede llegar a ser esta plataforma.

Para finalizar el desarrollo de la aplicación se han realizado algunas pruebas de funcionamiento, pruebas que efectivamente han sacado a la luz errores en la implementación o un mal funcionamiento. Todos estos errores han sido corregidos para ofrecer fiabilidad y robustez.

Por último, en lo referente al proyecto en general, se han anotado unas posibles mejoras y unas futuras actualizaciones que pueden ser de interés para el nicho de mercado al que se pretende acceder. Estas mejoras serán detalladas junto a las conclusiones finales.

2. Contexto tecnológico

2.1 Tecnología Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, de la misma forma que lo son iOS, Symbian y Blackberry OS. La principal diferencia, y bastante notable, es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma**.** El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El propio sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla y usando el conocido lenguaje de programación Java. Esta sencillez, unida a la existencia de IDE's (Entorno de Desarrollo Integrados) gratuitos como Eclipse, ha hecho posible el gran número de aplicaciones surgidas de unos pocos años para acá. De hecho este ha sido uno de los secretos del crecimiento de Android.

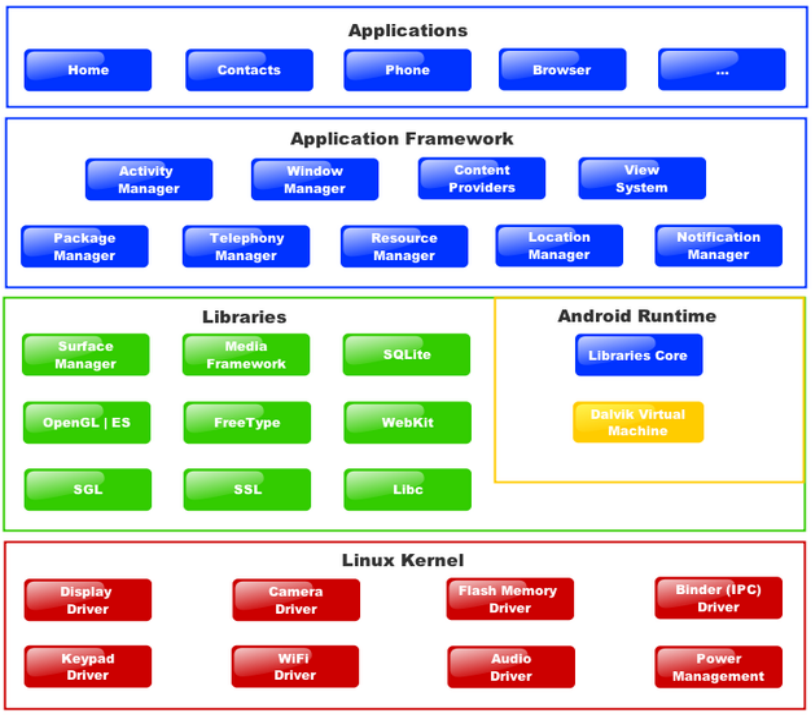


Ilustración 1 – Capas de Android

Por otra parte, como ya se ha reflejado, existe variedad de entornos de desarrollo con diferentes particularidades. En este caso concreto se ha optado por el IDE Eclipse, un IDE al que se le puede añadir un plugin de desarrollo Android para añadir funcionalidad. Concretamente ofrece el Android SDK Manager, el gestor de paquetes de Android, un gestor para descargar y actualizar las distintas API proporcionadas por Google. También ofrece la posibilidad de configurar tu propio dispositivo virtual mediante el AVDM (Android Virtual Device Manager). Con esta herramienta podemos personalizar un dispositivo virtual con distintos tamaños de pantalla, potencia, memoria, etc.

2.2 Android en el mercado

El sistema operativo para móviles, Android, ha tenido un crecimiento espectacular desde que surgiera el fenómeno de los *Smartphone*. Según un artículo del diario El País publicado en Julio de 2012 Android tenía en España una cuota de mercado del 84.1%, un incremento importantísimo si tenemos en cuenta que para el mismo mes del año 2011 la cuota de Android estaba en el 41.3%. El artículo, que se basa en el estudio de Kantar Worldpanel ComTech, también habla de importantes crecimientos en Europa, siendo Italia la cuota de mercado más baja con un notable 49.6%. Tal es así que en el año 2013 la cuota de mercado europea para Android se sitúa en un 70%.

Si tenemos en cuenta las versiones de Android el mercado está algo más fragmentado. Según los últimos datos las tres principales versiones de Android son 2.3 Gingerbread, 4.0 Ice Cream Sandwich y 4.1 Jelly Bean. Por tanto la aplicación contará como API base con la versión 2.3, permitiendo compatibilidad con prácticamente la totalidad de dispositivos del mercado. Cabe señalar que la siguiente tabla muestra simplemente una distribución desde la versión 2.2 hasta la versión 4.2.

Version	Codename	API	Distribution
2.2	Froyo	8	2.4%
2.3.3 - 2.3.7	Gingerbread	10	30.7%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	21.7%
4.1.x	Jelly Bean	16	36.6%
4.2.x		17	8.5%

Ilustración 2 - Distribución de versiones android

3. Análisis y diseño

El análisis de requisitos es una parte fundamental en toda aplicación, es la especificación de que debe hacer el programa, como debe hacerlo y quien puede hacerlo. En una hipotética aplicación para un cliente esta parte cobra aun más importancia si cabe. Así pues en este apartado se hará una reseña sobre cómo debe funcionar la aplicación.

3.1 Análisis de requisitos

Los principales requisitos de la aplicación son los siguientes:

- Listado de las diferentes monedas: La aplicación deberá mostrar un listado por países de las diferentes monedas.

- Almacenamiento permanente: La aplicación tendrá que almacenar de forma permanente de que monedas se dispone y de cuáles no.

- Listado de las monedas que se tienen: Se debe ofrecer al menos una forma de visualizar las monedas que ya forman parte de la colección.

- Visualización individual: Se debe añadir una opción para visualizar en grande e individualmente las monedas con el objetivo de poder identificarlas fácilmente.

- Borrar: Debe poderse borrar una moneda añadida a la colección. Esto principalmente obedece a la posibilidad de cometer errores al añadir.

- Interfaz y usabilidad: La interfaz debe ser simple e intuitiva. El objetivo es que la aplicación resulte práctica y cómoda, al menos más que apuntarlo en un papel.

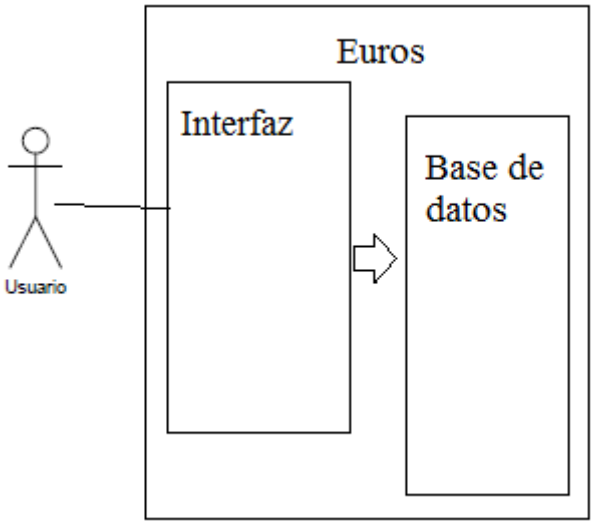
3.2 Requisitos técnicos

- Consistencia: La aplicación debe ser consistente mientras esté instalada en el teléfono. Por tanto, se debe emplear algún sistema de almacenamiento como bases de datos.

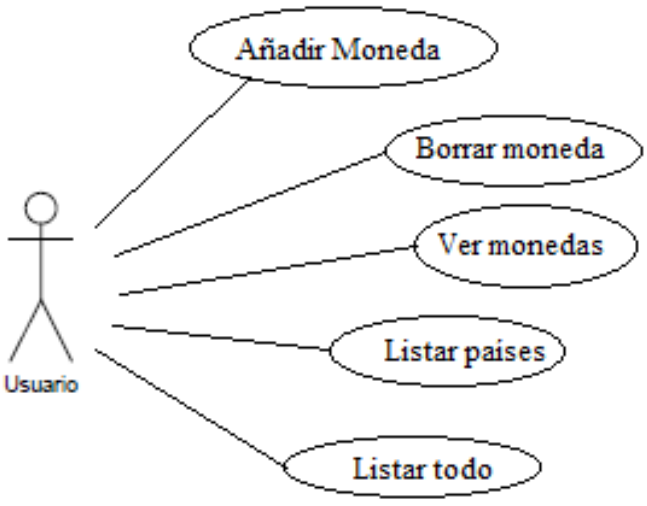
- Duplicidad: Debe evitarse la duplicidad de información para hacer un uso eficiente de la memoria, ya que algunos dispositivos disponen de poca.

3.3 Diagramas de casos de uso

- Diagrama de contexto



- Modelo de casos de uso



4. Implementación

4.1 Descripción del árbol del proyecto y componentes

En este apartado se va a explicar con mayor detalle el diseño y la implementación de la aplicación. Se hará también una explicación de los componentes utilizados y de su funcionamiento. Todo ello quedará ilustrado con fragmentos de código.

Como ya se ha comentado anteriormente se ha pretendido cubrir la mayor parte del mercado, por lo que la versión mínima de compilación ha sido la 2.2, esto garantiza que en cualquier dispositivo con una versión 2.2 de Android funcionará correctamente. Por otro lado también se han librerías de niveles superiores como la 4.0 para garantizar que los nuevos modelos de móvil funcionen igualmente bien.

El primer paso realizado a la hora de la implementación fue la toma de contacto con Android. Para ello se leyó diferente documentación, tanto libros como webs, y se realizaron pequeños programas para pruebas. Con esto se pretendía adquirir cierta soltura en el manejo de los diferentes componentes de Android.

Una vez adquirida esta mínima soltura se procedió a crear el propio proyecto en Eclipse, este proyecto finalmente tuvo la forma siguiente:

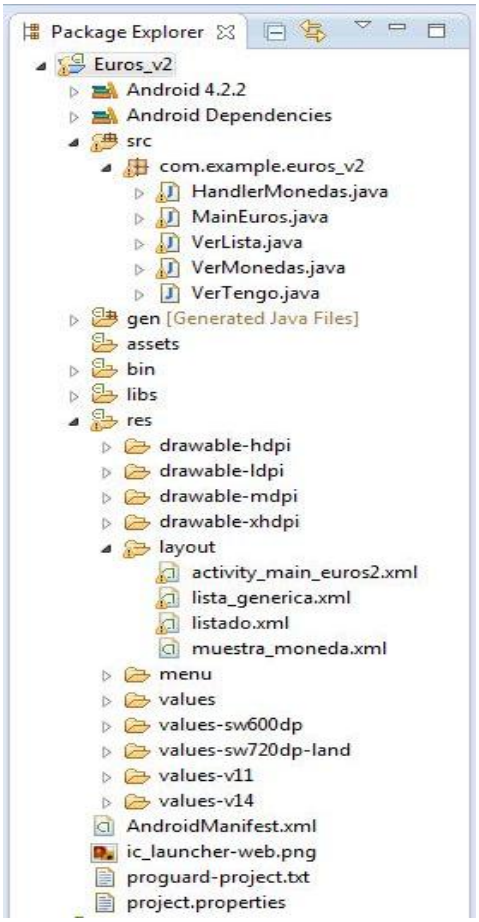


Ilustración 3 – Estructura del proyecto

Como se puede observar existen dos directorios principales. Uno es la carpeta "src", que contiene el *package* donde se encuentra el código Java/Android. Este directorio es el que almacena tanto la capa de aplicación como la capa de datos. La capa de aplicación son el *Main.java, VerLista.java, VerMonedas.java y VerTengo.java* las cuales contienen la llamada lógica de negocio. La clase *HandlerMonedas.java* es la clase encargada de la persistencia de datos. Ésta clase es la interficie de acceso a la base de datos y ofrece los controles oportunos para su manejo.

El otro directorio de relevancia para toda aplicación Android es el "res/layout". En este directorio se encuentran los archivos XML que configuran la interfaz gráfica de la aplicación, es decir, la capa de presentación. En este caso concreto se dispone de:

- El layout principal, el *activity_main_euros2.xml,* el layout que representa la pantalla de entrada a la aplicación.

- El layout *lista_generica.xml*. Este archivo dibuja una lista genérica a rellenar, esta lista contendrá todas las monedas que se tienen ordenadas por país y valor.

- El layout *listado.xml*. Esta pantalla es la encargada de mostrar las monedas por países. Contiene un desplegable llamado *spinner* para seleccionar el país, una vez seleccionado se dibujan las diferentes monedas del país y se indica si están en la base de datos o no. Esta será la forma típica de visualización puesto que ofrece la posibilidad de identificar la moneda.

- El layout *muestra_moneda.xml*. Este último layout ofrece la posibilidad de ver una a una las diferentes monedas de la zona euro. El layout consta de una imagen en grande de la moneda, dos intuitivas flechas de que simbolizan "anterior" y "siguiente", una cabecera que muestra el país y valor de la moneda y un desplegable en la parte inferior para saltar directamente a un país concreto.

Si bien estos son los principales directorios a tener en cuenta en toda aplicación, existen otros que también son de especial interés y que de hecho lo son para esta aplicación. Estos son los directorios "drawable". Como se puede observar existen diferentes directorios con dicho nombre, esto obedece a las diferentes configuraciones de resolución. El directorio "drawable" está pensado para albergar los recursos de imagen de la aplicación, tal es el caso que en este proyecto estos directorios contienen las imágenes de las monedas que se utilizaran, la imagen de fondo de la aplicación y las flechas de los botones. Las diferentes resoluciones que se nombran son:

- XHDPI significa Xtra-High-dpi, extra-alta resolución.

- HDPI significa High-dpi, alta resolución.

- MDPI significa Medium-dpi, media resolución.

- LDPI significa Low-dpi, baja resolución.

Según las necesidades particulares de cada proyecto o dispositivo para el que se pretenda programar habrá que colocar los recursos de imagen en distinta ubicación.

La última parte importante de la estructura del proyecto corresponde al archivo *AndroidMainfest.xml*. Este archivo contiene la configuración básica de la aplicación. Aquí es donde se puede configurar que versión del SDK utilizar, permisos especiales como conexión a Internet o GPS o por ejemplo decidir si la ventana de la aplicación podrá rotar si se gira el móvil.

Una vez vista le estructura principal de un proyecto Android y para qué sirven las partes principales se hará una breve descripción de los componentes utilizados, unos componentes básicos y versátiles.

- Layout: Es el elemento gráfico base de Android. Existen diferentes modelos de Layout según sus propiedades. Por ejemplo existe un LinearLayout que te alinea automáticamente los elementos, de forma configurable por su puesto. Existe también un RelativeLayout, el cual ofrece mucha más libertad a la hora de editar. Existen también GridLayout, TableLayout, etc.

- ScrollView: Es una vista particular puesto que debe cumplir ciertos requisitos, por ejemplo que solo tenga un padre (Layout) directo. Este elemento te permite crear un cuadro que tiene barra de *scroll*, algo muy necesario cuando la información a mostrar ocupa mucho.

- TextView: Es un campo de texto simple pensado para mostrar texto plano. Es un campo típico para enunciados o listas simples.

- EditText: Es un campo similar al TextView con la particularidad que es editable. Esto ofrece nuevas posibilidades de configuración como por ejemplo decidir si es numérico, numérico decimal, texto, etc.

- Spinner: Como ya se ha comentado de pasada, el Spinner es un desplegable el cual muestra el contenido de, por ejemplo, un *array de strings*. En este caso particular es utilizado para mostrar los diferentes países o valores de moneda.

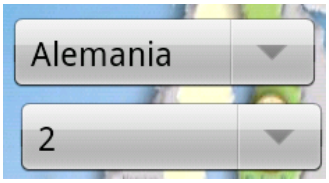


Ilustración 4 - Spinners

- CheckBox: Como su propio nombre indica es una caja de *cheked*. Esta caja ofrece la opción de marcar y desmarcar, algo útil a modo de confirmación o para evitar errores involuntarios.

Ilustración 5 - Checked Box

- Button: Otro elemento básico e intuitivo, un botón. El botón es altamente editable en cuanto a contenido, pudiéndose poner texto o cualquier imagen.

El botón será típicamente utilizado para lanzar nuevas pantallas o realizar funciones, por ejemplo añadir/borrar una moneda o visualizarlas.

- TableRow: Este elemento representa una tabla. Dicha tabla puede contener una considerable variedad de elementos, por ejemplo botones, cajas de textos o en este caso imágenes. En el caso particular de la TableRow existen diferentes versiones con funcionalidad más específica. Por ejemplo existe un TableLayout que fue una mejora añadida y que algunas API de Google no soporta.

Si bien estos elementos son básicos y aparecen en multitud de aplicaciones, existen otros elementos no utilizados en este proyecto. Por ejemplo Android dispone de botones más específicos, diferentes timers, VideoView, ImageView y otros muchos.

Es de interés señalar que todos estos elementos son editables tanto de forma estática, mediante los archivos XML, como de forma dinámica mediante código. Esto implica una mayor versatilidad y un gran abanico de opciones.

4.1 Descripción del código fuente y métodos

Pese a que Android está basado en el lenguaje de programación Java, presenta una serie de particularidades propias de sus dispositivos finales. Por ello Google ofrece una serie de API y revisiones que son actualizadas cada cierto tiempo y adaptadas a las nuevas versiones de su sistema operativo. A continuación se señalan los elementos que se han encontrado de mayor interés:

- Activity: Si bien en el fondo son clases java, Android se basa en *activities*. Estas son clases padre de Android que ofrecen una funcionalidad específica. Generalmente una clase extiende de *Activity* y esta clase está vinculada a alguno de los layouts del XML. De esta forma típicamente al lanzar una *activity* el usuario visualiza una nueva pantalla que le ofrece nueva información o funcionalidad.

- Métodos de escucha: Existen varios métodos de escucha, pero en lo que concierne a este proyecto hay dos de relevancia. Uno de ellos es el método *onClickListener()* que implementa la interficie *OnClickListener*. Este método de escucha se lanza cuando el objeto que contiene el *.setOnClickListene()r* es presionado. Una vez es lanzado empieza a ejecutar el código que se encuentra definido en su método *onClick()*. Este método es especialmente utilizado en botones.

- El otro método de especial interés es el *onItemSelectedListener()*, un método particular del objeto *Spinner* que en la práctica realiza una función similar. Cuando el *Spinner* es pulsado se detecta que elemento ha sido el seleccionado, entonces se lanza este método de escucha y se procede a ejecutar el código contenido en su método *onItemSelected()*. Una diferencia respecto al *onClickListener()* es que el método *onItemSelectedListener()* posee por defecto un método para el caso en que no se selecciones ningún elemento, el *onNothingSelected()*. Dentro de este método también se puede añadir código a realizar en dicha situación.

```
spPaises.setOnItemSelectedListener(new OnItemSelectedListener(){
    public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id){
        PAIS = parentView.getSelectedItem().toString();
    }

    public void onNothingSelected(AdapterView<?> parentView){

    }
});

//Se almacena en una variable global el contenido de la moneda
spMonedas.setOnItemSelectedListener(new OnItemSelectedListener(){
    public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id){
        MONEDA = parentView.getSelectedItem().toString();
    }

    public void onNothingSelected(AdapterView<?> parentView){

    }
});

View botonAgregar = findViewById(R.id.buttonAgregar);
botonAgregar.setOnClickListener(new OnClickListener(){
    public void onClick (View v){
        CheckBox te = (CheckBox)findViewById(R.id.checkBoxTengo);
        String res = "Error, no has marcado el check";
        //Si está marcado el checked para insertar se inserta y se muestra un toast indicandolo, si no está marcada
        if(te.isChecked()){
            res = handler.addMoneda(PAIS.toString().toUpperCase().trim(),
                    MONEDA.toString().toUpperCase().trim(),te.getText().toString().trim());
            Toast toast = Toast.makeText(getApplicationContext(), res, Toast.LENGTH_SHORT);
            toast.show();
        } else {
            Toast toast = Toast.makeText(getApplicationContext(), res, Toast.LENGTH_SHORT);
            toast.show();
        }
        te.setChecked(false);
    }
});
```

Ilustración 6 – Métodos escucha

- Intent: Los intent son básicamente mecanismos que nos permiten llamar a aplicaciones externas a la nuestra, lanzar eventos a los que otras aplicaciones puedan responder, lanzar alarmas etc. Por lo tanto los intent son la forma de lanzar actividades nuevas, las actividades que realizaran nuevas funciones. Estos mecanismos incorporan dos funcionalidades especialmente interesantes. Una de estas es la capacidad de enviar datos, de forma que la nueva actividad puede recogerlos y trabajar con ellos. La otra capacidad interesante es la inversa, la de recoger los datos que le vienen de su actividad hijo para así actuar en base a ellos o manejarlos.

- SQLite: SQLite es una de las grandes armas de Android, es un pequeño gestor de bases de datos basado en MySQL integrado en la API de android. Pese a ser una versión simplificada de MySQL y pensada para móviles, SQLite ofrece una gran potencia y fiabilidad. Además de eso SQLite ofrece formas simples de manejo y métodos que te permiten ejecutar sentencias SQL de forma muy simple. Para operar con facilidad con SQLite típicamente se crea una clase manejadora (Handler) que se encargará de hacer de interficie entre capas. Este manejador creara y tendrá acceso directo a la base de datos mientras que las clases de la capa superior podrán llamar a esta clase para operar con la base de datos.

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.widget.Toast;


public class HandlerMonedas extends SQLiteOpenHelper {

public HandlerMonedas(Context ctx){

        super(ctx,"Monedas",null,1);

    }

    @Override
    public void onCreate(SQLiteDatabase db){
        //Creamos la tabla monedas
        String sql = "CREATE TABLE monedas(ID INTEGER PRIMARY KEY AUTOINCREMENT, pais TEXT, valor TEXT, tengo TEXT)";
        db.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int va, int vn){
        db.execSQL("DROP TABLE IF EXIST monedas");
        onCreate(db);
    }
```

Ilustración 8 - Handler

Estas son las características más importantes encontradas durante la realización del proyecto, al menos las más importantes que difieran de Java. Cabe recordar que Android está basado en Java y puede importar cualquiera de sus librerías y trabajar con ellas en cualquier momento, por lo que la programación Android resulta muy semeja a la programación Java.

5. Funcionamiento de la aplicación

5.1 Funcionalidad

Como ya se ha señalado, la funcionalidad básica de la aplicación es la propia para toda aplicación que pretenda ser un gestor persistente. Es decir, la aplicación debe mantener y mostrar información del estado de una colección personal de monedas. Descrito de una forma más esquemática y por funciones tenemos:

- Botón de añadir: El botón encargado de guardar en la base de datos que tenemos una nueva moneda. Previo a este paso debe confirmar con el checked que la queremos añadir para evitar errores. Este botón recogerá la información que hay en los desplegables.

- Botón de borrar: Es el botón encargado de borrar de la base de datos que tenemos cierta moneda. La información de que moneda debe borrar será recogida de los desplegables.

- Ver monedas: Esta es la opción que permite ver las monedas de forma independiente. En la pantalla para visualizar monedas se tienen diferentes elementos. En el centro y grande la moneda, dos flechas para pasar de monedas, una cabecera que indica la moneda y el valor correspondiente a la imagen y un desplegable de países para poder saltar directamente.

- Lista por país: Será la lista más utilizada puesto que ofrece una vista por países con un dibujo miniatura que permite identificar la moneda. En esta vista sale reflejado si la moneda ya forma parte de la colección o, si por el contrario, aun no está. Esta pantalla también cuenta con un desplegable por países para ir pasando de uno a otro.

- Lista que tengo: Una lista rápida en texto plano que muestra el contenido de la colección. Esta pretende ser una lista rápida para en un momento dado tener conocimiento de todo el contenido de la colección.

5.2 Pruebas y medidas de seguridad

Durante la implementación se han ido realizando diferentes pruebas de funcionamiento y de control de errores. Estas pruebas tienen que ver en su mayoría con el acceso a base de datos. Ha sido necesario comprobar repetidas veces que la aplicación insertaba correctamente en la base de datos, de forma que no faltaran ni sobrasen elementos. También se ha pretendido evitar duplicidades y no poder ingresar repetidas veces una misma moneda. Por motivos de eficiencia en la base de datos solo constan las monedas en posesión del usuario, es decir, las monedas que no tiene el usuario se deducen, dado que se conoce previamente el número de monedas y su forma.

Con respecto al borrado se ha tenido cuidado de no borrar por completo la base de datos, por lo tanto se ha identificado claramente las distintas monedas y se han establecido controles.

También ha sido necesario controlar los datos insertados en la base de datos. La idea es que el usuario no pueda introducir datos erróneos como, por ejemplo, un valor o país inexistente. Una forma fácil de establecer estos controles ha sido el uso de desplegables. Los desplegables ofrecen unas opciones limitadas a la hora de seleccionar, unas opciones establecidas en programación que dejan al usuario solo frente a información válida. Estos elementos presentan además la ventaja añadida que reduce el código, puesto que no es necesario establecer funciones que controlen la información.

6. Publicación en el Market

Google, de la misma manera que otros sistemas operativos móviles, dispone de un market propio para ofertar aplicaciones y utilidades. Este market de Google es llamado Google Play. En lo referente a este trabajo será de de interés conocer la forma de subir una aplicación propia a este market para que los usuarios de Android dispongan de ella.

El primer requisito, el más obvio, y no por ello menos importante es tener cuenta de Google. Una vez se tenga una cuenta de Google hay que dirigirse a esta dirección: http://market.android.com/publish donde se indicaran los pasos a seguir.

Previo a este paso tal vez resulte de interés mencionar algunas cosas más. Se debe tener en cuenta que no es totalmente gratis publicar una aplicación en Google Play, esto quiere decir que para registrarse como desarrollador hay que pagar una cuota. Esta cuenta son 25 USD (Dólares americanos) y es una cuota de por vida, por lo tanto no siguiendo la filosofía *libre* es un desembolso bastante asequible. Una vez realizado el pago (que puede hacerse por distintos canales, entre ellos el propio Google Wallet, antiguo Google Checkout) ya se es desarrollador Google, por lo tanto ya se pueden subir aplicaciones.

Para subir una aplicación a Google Play se necesita el fichero instalador de Android, el fichero de extensión *.apk*. Este fichero puede ser generado con el propio eclipse yendo a su opción *file/export/Export Android Application*. Siguiendo los sencillos pasos que aparecen se obtiene un archivo .apk que puede ser instalado en cualquier dispositivo Android.

Ya habiendo pagado la licencia y teniendo el archivo *.apk* hay que dirigirse a la siguiente dirección: https://market.android.com/publish/Home donde aparece un enlace "Subir aplicación" para proceder a la subida del fichero *.apk*. Hay que señalar que Google te pide una serie de datos mínimos a la hora de subir una aplicación, estos datos están enfocados a definir y explicar el funcionamiento de ésta para que los usuarios sean conscientes de que se van a descargar.

Los datos necesarios son:

- Mínimo dos capturas de pantalla de la aplicación: es recomendable que tengan buena calidad pues aparecerán cuando el usuario pulse en "Más" en Google Play.
- Icono de la aplicación: la aplicación debe identificarse con un icono, a ser posible realizado por nosotros mismos pues éste aparecerá en la parte izquierda cuando los usuarios busquen y encuentren nuestra aplicación en Google Play.
- Opcionalmente podemos incluir una imagen promocional.
- Opcionalmente podemos incluir una imagen de funciones.
- Opcionalmente podremos incluir un vídeo promocional de YouTube.
- Si no queremos que la aplicación sea promocionada fuera de Google Play hay que marcar el check: "No promocionar mi aplicación salvo en Google Play y en los sitios web o para móviles propiedad de Google. Asimismo, soy consciente de que cualquier cambio relacionado con esta preferencia puede tardar sesenta días en aplicarse".

- Podremos elegir varios idiomas para añadir la descripción de las funciones y uso de nuestra aplicación. El inglés es obligatorio. En este punto nos solicitará:
    - Título de la aplicación: será el nombre que aparezca en las búsquedas, no debe ser muy largo (inferior a 30 caracteres).
    - Descripción: una descripción detallada (hasta 4000 caracteres) de lo que hace la aplicación, es aquí donde hay que convencer al usuario de que nuestra aplicación tiene funciones únicas.
    - Cambios recientes: si es una actualización, se puede indicar aquí las últimas mejoras de la aplicación.
    - Si se ha añadido un vídeo promocional, podemos añadir un texto promocional.
    - Tipo de aplicación: seleccionar del desplegable el que más se ajuste.
    - Categoría: seleccionar del desplegable la que más se ajuste.
- Clasificación del contenido: marcar si la aplicación es para todos los públicos o contiene algún tipo de contenido para mayores.
- Precios: indicar aquí si la aplicación será gratuita o de pago.
- Si se ha elegido de pago, en "Precio predeterminado" se debe introducir el precio que se considere que ha de tener la aplicación. Pulsando el botón "Autocompletar" hará los ajustes para los diferentes países en los que se publique.
- También se indicará el número aproximado de modelos de dispositivos Android que soportarán la aplicación según los filtros indicados en el archivo del *mainfest*. Esto es algo que ya se explicó a la hora de elegir la cuota de mercado.
- Por último introducir la información de contacto:
    - Sitio web.
    - Correo electrónico.
    - Teléfono.

Con todo esto la aplicación ya estará publicada en el market de Google y la gente tendrá acceso a ella. Comentar también que en caso de que la aplicación sea de pago es conveniente vincular a la cuenta de Google con la de Google Wallet de forma que los cobros lleguen sin problemas.

7. Conclusiones

7.1 Futuras mejoras

Una vez concluida la aplicación y testeado su funcionamiento se pensaron algunas mejoras que podrían resultar interesantes. Estas mejoras pueden ser futuras actualizaciones o incluso versiones de pago a modo de versión ampliada.

Una de estas mejoras tiene que ver con un tipo de moneda no contemplada, la moneda conmemorativa. Dado que cada país tienen la capacidad de emitir monedas de 2€ con motivos especiales despiertan gran interés entre los coleccionistas. Por tanto, es interesante que la aplicación contemple estas monedas y realice una gestión de las mismas.

Otra posible mejora tiene que ver con la actualización del catálogo. Como se ha comentado existen monedas conmemorativas, estas monedas son diferentes según el motivo y el año de emisión, por lo que es necesario que se compruebe que el catálogo que se posee es correcto. Lo mismo ocurre con otros países como por ejemplo el Vaticano, un país que en función del Papa emite una moneda diferente. Y por último, contemplar la posibilidad de que nuevos países se incorporen a la zona monetaria. Si bien esto puede ser manejado en forma de actualizaciones periódicas, resulta interesante ofrecer la posibilidad de automatizarlo.

En cuanto al diseño se ha pensado en mejorar la visualización. Dado que son países y valores resulta intuitivo dibujar una tabla en que se crucen países y valores y marcar de alguna forma las monedas que se tienen, esta forma parece algo más intuitiva para el usuario que el actual sistema.

7.2 Conclusiones finales

Cuando se empezó el proyecto los conocimientos de Android eran más bien nulos, el objetivo principal fue iniciarse en la programación en Android, conocer su funcionamiento básico y ser capaces de desarrollar completamente una sencilla aplicación.

Para conseguir estos objetivos ha sido necesario documentarse sobre Android mediante diferentes canales y realizar multitud de código de “prueba y error”. Finalmente se consiguió adquirir cierta soltura en la programación y se ha visto cierto progreso.

En el momento de dar por cerrado el proyecto la aplicación presenta una funcionalidad aceptable que, pese a ser mínima, sirve eficientemente para su propósito. El gestor permite mantener una fiable lista de las monedas que se tienen y gestionarlas.

No obstante, es cierto que la aplicación necesita mejoras, algunas de ellas comentadas en el apartado anterior. Para pretender su lanzamiento al mercado es necesario aportar algo más de valor al usuario, bien sea en funcionalidad o diseño.

Por lo tanto, puede decirse que el proyecto la alcanzado sus objetivos mínimos satisfactoriamente. Se ha introducido la programación en Android de forma básica

consiguiendo una aplicación satisfactoria, ofreciendo nuevas mejoras y una base de conocimiento para el futuro. También se ha podido determinar que el mercado de Android es un mercado en expansión que presenta un gran número de oportunidades para gente nueva, y que resulta muy atractivo debido a que los costes del proyecto pueden ser considerablemente reducidos.

7. Bibliografía

Manuales

- Curso de programación Android de Salvador Gómez Oliver.
- Android: guía para desarrolladores. De W. Frank Ableson, Robi Sen, Chris King.

Recursos multimedia y web

- API de Google para desarrollar en Android.
  http://developer.android.com/sdk/index.html
- API de Java.
  http://docs.oracle.com/javase/7/docs/api/
- Curso de programación de aplicaciones en Android, Polimedia - UPV.
  Jesús Tomás Gironés.
  https://polimedia.upv.es/catalogo/mobile/curso.asp?curso=9e0e4ba0-d852-2243-8590-320e9d78ff36
- Curso básico Android.
  http://www.aprendeandroid.com/menu.htm
- Artículo del diario El País.
  http://tecnologia.elpais.com/tecnologia/2012/07/12/actualidad/1342106677_887889.html
- Vídeo-tutoriales de YouTube
  http://www.youtube.com/user/edu4java/search?query=android
- Diversos artículos.
  www.elandroidlibre.com
  www.xatakandroi.com
  http://elbauldelprogramador.com/category/programacion/android/
- Publicación en Google Play
  https://support.google.com/googleplay/android-developer/answer/113469?hl=es
  http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=561