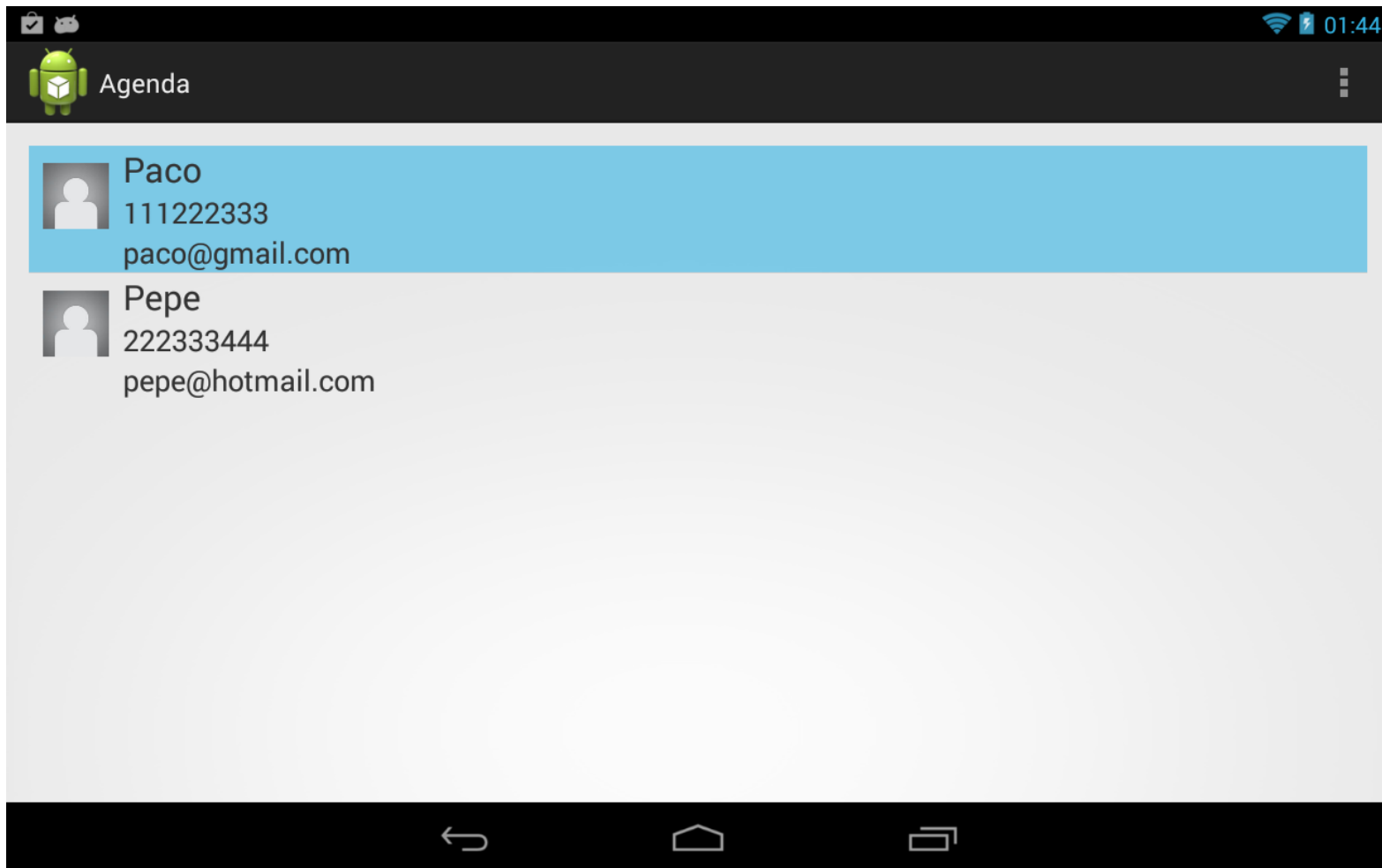


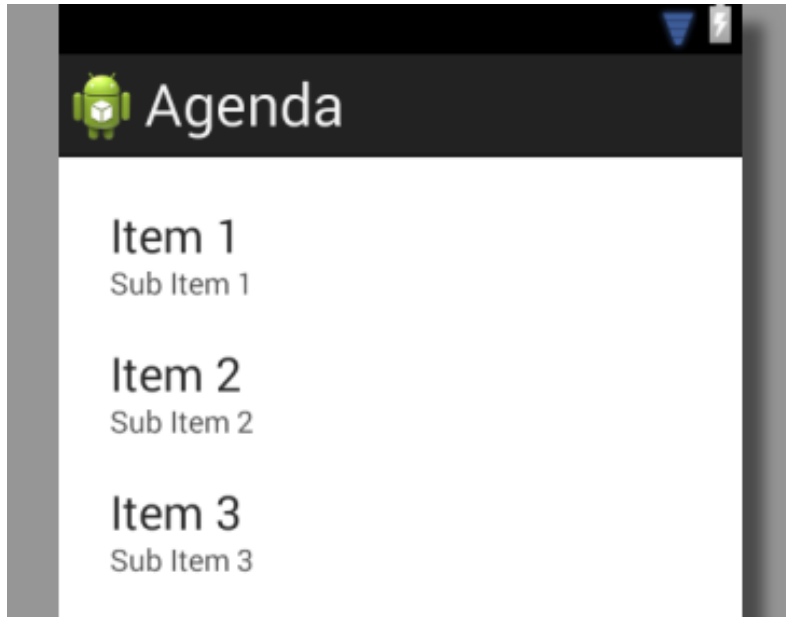
# **IMPLEMENTACIÓN DE UNA AGENDA**

**Acceso a datos**

# APLICACIÓN AGENDA



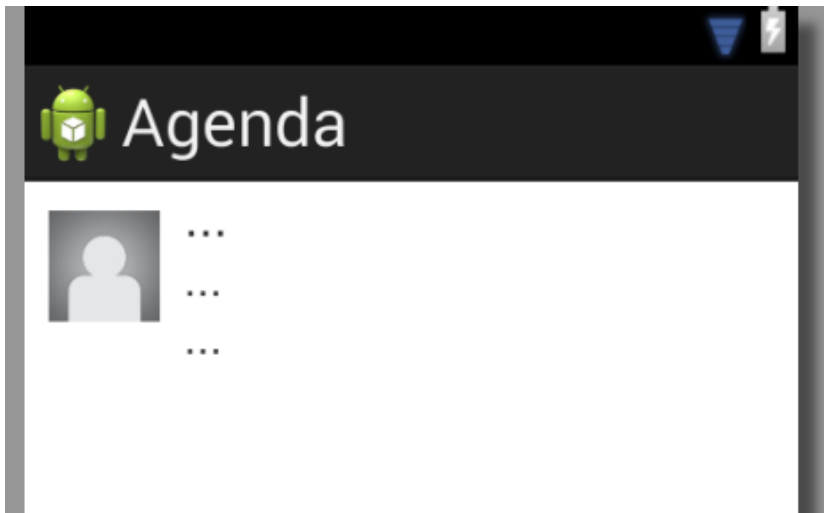
# LAYOUT PRINCIPAL



```
<LinearLayout>  
<ListView  
    android:id="@+id/lvLista">  
</ListView>  
</LinearLayout>
```



# LAYOUT DETALLE



```
<LinearLayout
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/ivUsuario"/>
    <LinearLayout
        android:orientation="vertical" >
        <TextView
            android:id="@+id/tvUsuario"/>
        <TextView
            android:id="@+id/tvTelefono"/>
        <TextView
            android:id="@+id/tvCorreo"/>
    </LinearLayout>
</LinearLayout>
```



## STRINGS.XML

- ```
<resources>
  <string name="app_name">Agenda</string>
  ...
  <string-array name="/lst_usuarios">
    <item>Paco</item>
    <item>Pepe</item>
  </string-array>
  <string-array name="/lst_correos">
    <item>paco@gmail.com</item>
    <item>pepe@hotmail.com</item>
  </string-array>
  <string-array name="/lst_telefonos">
    <item>111222333</item>
    <item>222333444</item>
  </string-array>
</resources>
```



# CLASE ADAPTADOR

- Una clase adaptador hace de puente entre los datos que se deben visualizar y el ListView.
- Al constructor de la clase adaptador se le debe pasar el Contexto en el que se va a mostrar los elementos del ListView y los datos que se van a mostrar.
- Se debe sobrescribir el método **public View getView(int posicion, View vista, ViewGroup padre)** de la clase adaptador.
- La clase adaptador extiende la clase ArrayAdapter. Existen otras clases adaptadoras para otros tipos de datos.



# EL MÉTODO GETVIEW()

- El método

**public View getView (int posicion, View vista,  
ViewGroup padre)**

es invocado por Android al tratar de dibujar el ListView.

- Puesto que la clase conoce el número de elementos de la lista, y el usuario puede visualizarla subiendo y bajando los elementos, Android siempre mostrará un rango de elementos de la lista.
- Para mostrar un elemento en concreto se pasan como argumentos el número de elemento, la vista en la que se debe mostrar y el contenedor de la vista.



# IMPLEMENTACIÓN DE LA CLASE ADAPTADOR

```
public class AdaptadorBasico extends ArrayAdapter<String> {  
    private Context contexto; private String[] lista;  
    public AdaptadorBasico(Context contexto, String[] lista) {  
        super(contexto, R.layout.layout_detalle, lista);  
        this.contexto = contexto;  
        this.lista = lista;  
    }  
    @Override  
    public View getView(int posicion, View vista, ViewGroup padre) {  
        LayoutInflater i = (LayoutInflater) contexto.getSystemService  
            (Context.LAYOUT_INFLATER_SERVICE);  
        vista = i.inflate(R.layout.layout_detalle, null);  
        TextView tv = (TextView) vista.findViewById(R.id.tvUsuario);  
        tv.setText(lista[posicion]);  
        return vista;  
    }  
}
```





## CLASE PRINCIPAL

@Override

```
protected void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.layout_principal);  
    String[] datos = getResources().getStringArray(  
  R.array.lst_usuarios);  
    ListView lv = (ListView)findViewById(R.id.lvLista);  
    lv.setAdapter(new AdaptadorBasico(this, datos));  
}
```



## MEJORAR LA IMPLEMENTACIÓN

- Para mostrar un contacto completo – el nombre, el teléfono y el correo electrónico – se necesita proporcionar todos los datos a la clase adaptadora.
- En lugar de implementar una clase que extiende un `ArrayAdapter<String>` vamos a implementar una clase que extienda un `ArrayAdapter<Contacto>`.
- Debemos crear en primer lugar la clase contacto.



# CLASE CONTACTO

- public class Contacto {  
    private String nombre, telefono, correo;  
    public Contacto(String nombre, String telefono,  
                    String correo) {...}  
    public String getNombre() {...}  
    public void setNombre(String nombre) {...}  
    public String getTelefono() {...}  
    public void setTelefono(String telefono) {...}  
    public String getCorreo() {...}  
    public void setCorreo(String correo) {...}  
}



# NUEVA CLASE ADAPTADOR

```
public class Adaptador extends ArrayAdapter<Contacto> {  
    private Context contexto;  
    private ArrayList<Contacto> lista;  
    public Adaptador(Context contexto, ArrayList<Contacto> lista) {  
        super(contexto, R.layout.layout_detalle, lista);  
        this.contexto = contexto;  
        this.lista = lista;  
    }  
    @Override  
    public View getView(int posicion, View vista, ViewGroup padre) {  
        LayoutInflater i = (LayoutInflater) contexto.getSystemService  
            (Context.LAYOUT_INFLATER_SERVICE);  
        vista = i.inflate(R.layout.layout_detalle, null);  
        Contacto ct = lista.get(posicion);  
        TextView tv = (TextView) vista.findViewById(R.id.tvUsuario);  
        tv.setText(ct.getNombre()); ...  
        return vista;  
    }  
}
```



# NUEVA CLASE PRINCIPAL I

@Override

```
protected void onCreate(Bundle savedInstanceState){
```

```
...
```

```
String[] datos0 = getResources().getStringArray(  
  R.array.lst_usuarios);
```

```
String[] datos1 = getResources().getStringArray(  
  R.array.lst_telefonos);
```

```
String[] datos2 = getResources().getStringArray(  
  R.array.lst_correos);
```



## NUEVA CLASE PRINCIPAL II

```
ArrayList<Contacto> al = new ArrayList<Contacto>();  
for (int i = 0; i < datos0.length; i++) {  
    Contacto c = new Contacto(datos0[i], datos1[i],  
                                datos2[i]);  
    al.add(c);  
}  
Adaptador adap = new Adaptador(this, al);  
ListView lv = (ListView) findViewById(R.id.lvLista);  
lv.setAdapter(adap);  
}
```



## MEJORAR EL MÉTODO GETVIEW()

- Se puede mejorar el método **getView()**. Android permite reutilizar la vista de detalle. Cuando se está construyendo la lista se deben construir todas las vistas de detalle que se muestran. Pero al desplazarse por la lista hacía arriba y abajo, en lugar de volver a crear la vista (`inflate()`) se puede aprovechar el parámetro `View` del método.
- ```
if (vista == null) {  
    LayoutInflater inflater = (LayoutInflater)contexto.  
        getSystemService(  
            Context.LAYOUT_INFLATER_SERVICE);  
    vista = inflater.inflate(R.layout.lyt_detalle, null);  
}
```



## EJERCICIO

- Implementa la agenda usando el AdaptadorBasico.
- Termina de implementar la agenda utilizando para ello todas las mejoras propuestas.
- Modifica el archivo strings.xml para que contenga inicialmente quince contactos.
- Implementa alguna forma de añadir nuevos contactos desde la aplicación.

