

Android Avanzado



ANDROID

¿Qué vamos a ver?

- Patrones de diseño y navegación
 - ViewPager
 - Tabs
 - Navigation Drawer
 - Navegación vía el ActionBar

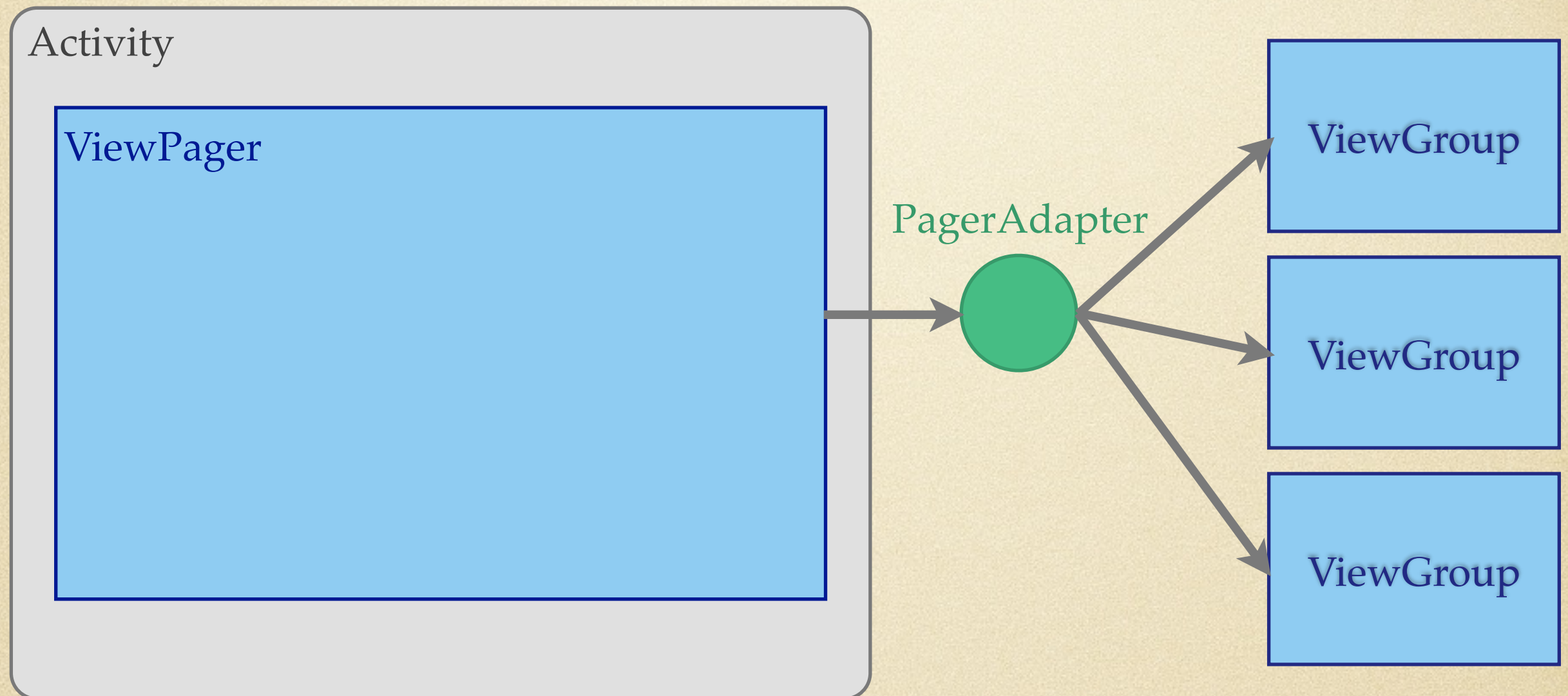
ViewPager

- Layout manager que permite mostrar y recorrer vistas como si fueran páginas
- Necesita un PagerAdapter
- Generalmente usado con Fragments
 - FragmentPagerAdapter y
FragmentStatePagerAdapter
- Provee animaciones por defecto

ViewPager

- Se utiliza para implementar:
 - galerías de imágenes / diapositivas
 - asistentes / wizards
 - tabs (en conjunto con ActionBar)

ViewPager



Implementación

- Creamos los Fragments que compondrán las páginas del ViewPager y sus layouts respectivos:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/content" >
    <TextView android:id="@+id/tviTextoSliide"
        style="?android:textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</ScrollView>
```

```
public class ScreenSlidePageFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(
            R.layout.fragment_screen_slide_page, container, false);
        // Cambiamos el texto del TextView para mostrar la página
        TextView tviTextoSliide = (TextView) rootView.findViewById(R.id.tviTextoSliide);
        tviTextoSliide.setText(String.format("Página %d", numPagina));
        return rootView;
    }
}
```

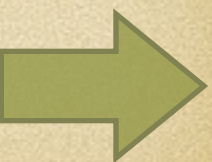

Implementación

- Creamos el layout del Activity que contendrá el ViewPager:

```
<android.support.v4.view.ViewPager xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</android.support.v4.view.ViewPager>
```

- Creamos el Activity que contendrá el ViewPager. Este activity crea nuestra implementación de PagerAdapter y lo conecta al ViewPager:



Implementación

```
public class ViewPagerSlidesActivity extends FragmentActivity {

    // Cantidad de páginas a mostrar
    private final static int NUMERO_PAGINAS = 5;

    private ViewPager viewPager;
    private PagerAdapter pagerAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_pager_slides);

        // Aquí creamos el PagerAdapter y lo asignamos al ViewPager
        viewPager = (ViewPager) findViewById(R.id.pager);
        pagerAdapter = new ScreenSlidePagerAdapter(getSupportFragmentManager());
        viewPager.setAdapter(pagerAdapter);
    }

    private class ScreenSlidePagerAdapter extends FragmentStatePagerAdapter {

        public ScreenSlidePagerAdapter(FragmentManager fm) {
            // FragmentStatePagerAdapter no implementa un constructor por defecto
            // Por eso necesitamos implementar este
            super(fm);
        }

        @Override
        public Fragment getItem(int position) {
            return ScreenSlidePageFragment.crear(position);
        }

        @Override
        public int getCount() {
            return NUMERO_PAGINAS;
        }
    }
}
```


Tabs

- Se implementan en el ActionBar
 - Permite mostrarlos en diferentes lugares de acuerdo al tamaño de la pantalla
- Se utilizan las clases `ActionBar.Tab` y `ActionBar.TabListener`
 - `TabListener` sólo indica cuál tab fue seleccionado.

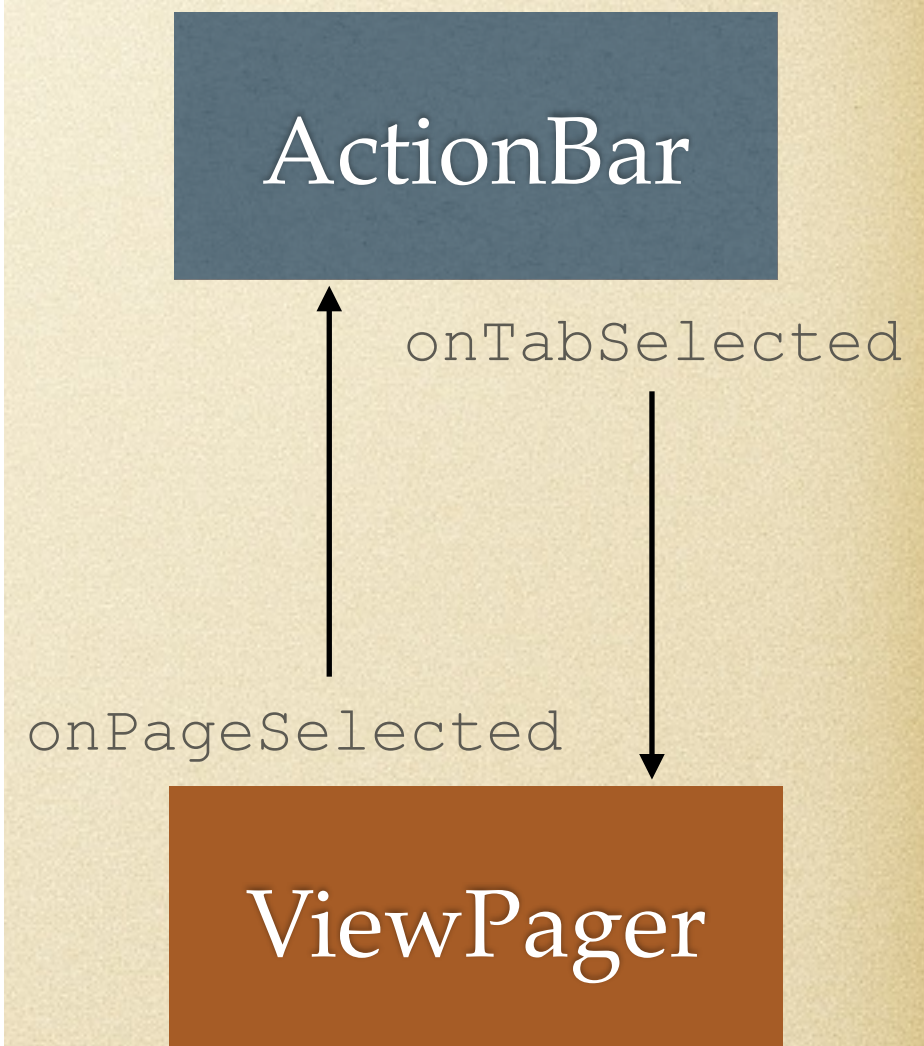
Tabs - Implementación

```
// Configuramos el ActionBar para que muestre tabs
final ActionBar actionBar = getActionBar();
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
// Creamos el TabListener para recibir los eventos de cambio de tab
ActionBar.TabListener tabListener = new ActionBar.TabListener() {
    @Override
    public void onTabSelected(Tab tab, FragmentTransaction ft) {
        viewPager.setCurrentItem(tab.getPosition(), false);
    }

    @Override
    public void onTabUnselected(Tab tab, FragmentTransaction ft) {
    }

    @Override
    public void onTabReselected(Tab tab, FragmentTransaction ft) {
    }
};
// Creamos los tabs
for (int i = 0; i < NUMERO_PAGINAS; i++) {
    actionBar.addTab(actionBar.newTab().setText("Tab " + i)
        .setTabListener(tabListener));
}

// Con esto, actualizamos el tab bar cuando se hace swipe
viewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener(){
    @Override
    public void onPageSelected(int position) {
        actionBar.setSelectedNavigationItem(position);
    }
});
});
```



Tabs - Implementación

- Alternativa para tabs 'ligeros': utilizar PagerTitleStrip / PagerTabStrip
- Necesario haber definido el título de cada tab en el PagerAdapter

```
<android.support.v4.view.ViewPager xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewPagerSlidesActivity">

    <android.support.v4.view.PagerTabStrip
        android:id="@+id/pager_title_strip"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:background="#33b5e5"
        android:textColor="#fff"
        android:paddingTop="4dp"
        android:paddingBottom="4dp" />

</android.support.v4.view.ViewPager>
```


Navigation Drawer

- Componente que permite un rápido acceso a los puntos importantes / lugares frecuentes de la aplicación
- Accesible desde cualquier pantalla de la aplicación
- Para implementarlo: `DrawerLayout` + `ListView` + `Fragments`

Navigation Drawer - Implementación

- Creamos el DrawerLayout en el Activity principal

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- Aquí va el contenido principal -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <!-- El Navigation Drawer -->
    <ListView android:id="@+id/left_drawer"
        android:layout_width="240dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="0dp"
        android:background="#111" />
</android.support.v4.widget.DrawerLayout>
```


Navigation Drawer - Implementación

- El menú lateral se implementa con un `ListView`

```
public class MainActivity extends Activity {  
    private String[] mPlanetTitles;  
    private DrawerLayout mDrawerLayout;  
    private ListView mDrawerList;  
    ...  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mPlanetTitles = getResources().getStringArray(R.array.planets_array);  
        mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);  
        mDrawerList = (ListView) findViewById(R.id.left_drawer);  
  
        // Setear el Adapter para el ListView  
        mDrawerList.setAdapter(new ArrayAdapter<String>(this,  
            R.layout.drawer_list_item, mPlanetTitles));  
        // Setear el listener para el ListView  
        mDrawerList.setOnItemClickListener(new DrawerItemClickListener());  
  
        ...  
    }  
}
```


Navigation Drawer - Implementación

- Para implementar las acciones del menú, se utiliza el `OnItemClickListener` del `ListView`

```
private class DrawerItemClickListener implements ListView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView parent, View view, int position, long id) {
        selectItem(position);
    }
}

/** Intercambia fragments en la vista de contenido principal */
private void selectItem(int position) {
    // Crear un nuevo Fragment y especificar el contenido basado en la posición
    Fragment fragment = new PlanetFragment();
    Bundle args = new Bundle();
    args.putInt(PlanetFragment.ARG_PLANET_NUMBER, position);
    fragment.setArguments(args);

    // Reemplazar el Fragment en el contenido
    FragmentManager fragmentManager = getFragmentManager();
    fragmentManager.beginTransaction()
        .replace(R.id.content_frame, fragment)
        .commit();

    // Resaltar el item seleccionado, actualizar el título, y cerrar el Drawer
    mDrawerList.setItemChecked(position, true);
    setTitle(mPlanetTitles[position]);
    mDrawerLayout.closeDrawer(mDrawerList);
}
```