Ejemplo con ViewPager en Android

- Pasos para la creación del primer ejemplo del taller.

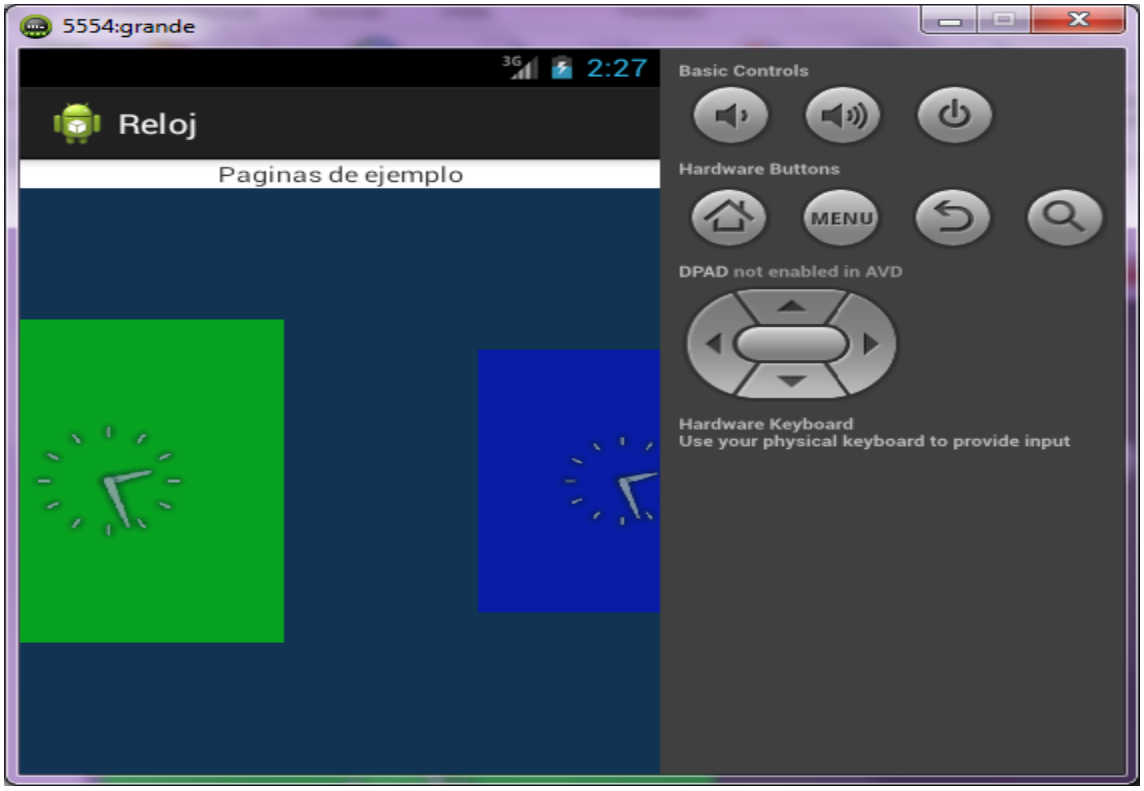
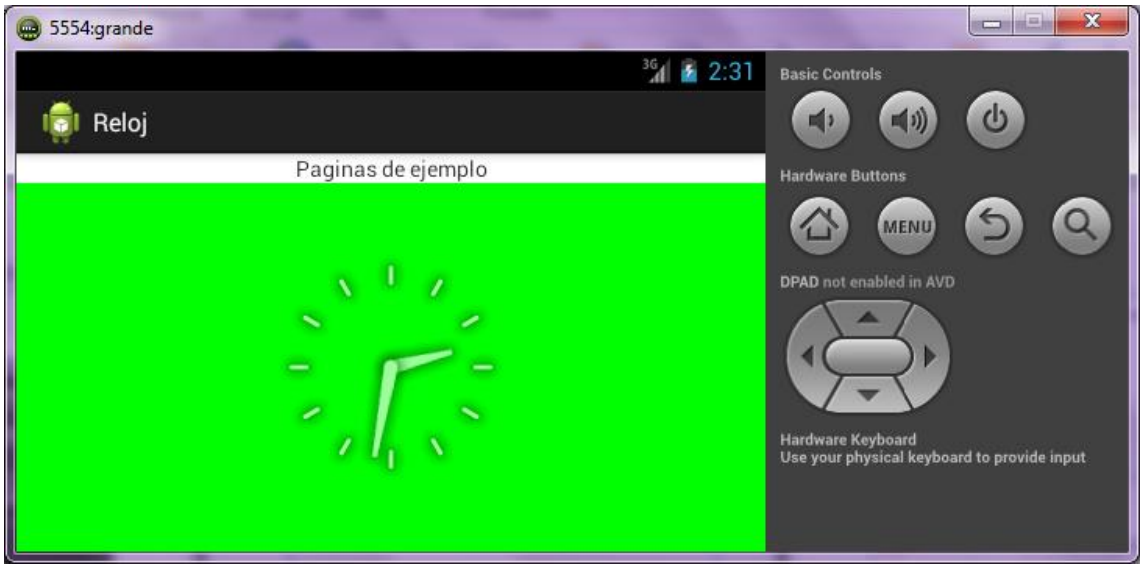
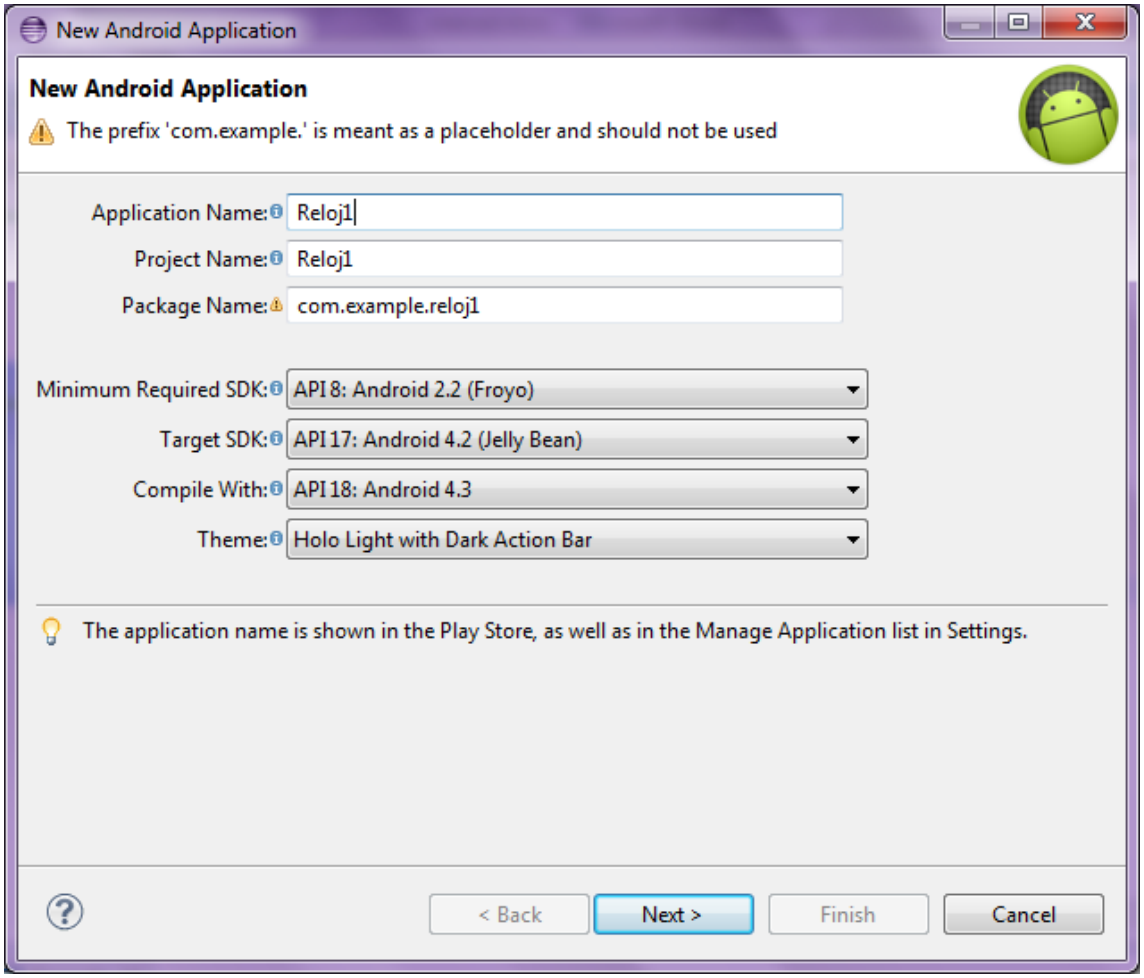


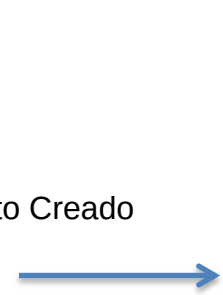
Ilustración 1 Portrait(Vertical)

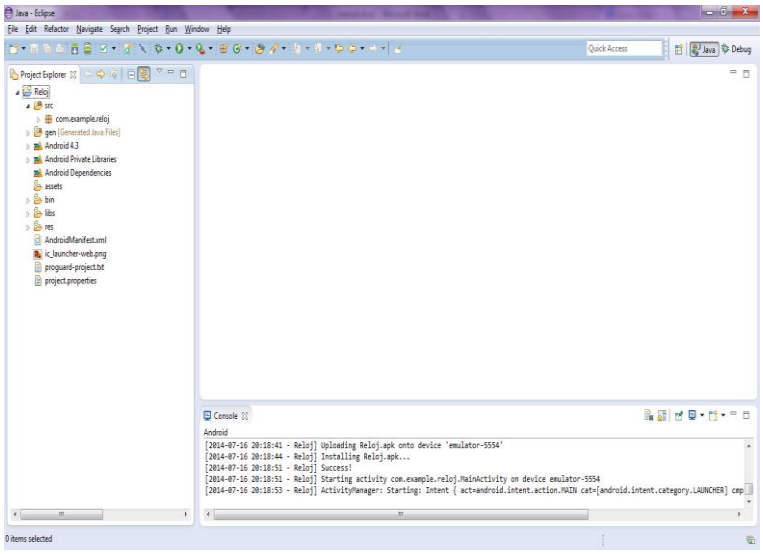


1. Creamos el proyecto
    a. Definimos el nombre del proyecto a realizar
    b. Podemos cambiar el package name.
    c. Selección del minimo de SDK y el target SDK



Proyecto Creado





2. Una vez creado el proyecto, iniciaremos con la definición de los layouts que necesitaremos para el ejemplo.
    a. Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Paginas de ejemplo" />

    <android.support.v4.view.ViewPager
        android:id="@+id/pager"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="#123456" />

</LinearLayout>
```

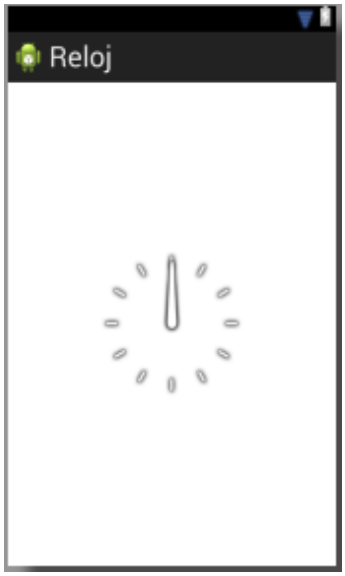
Ilustración 2 Resultado Main.xml

    b. Reloj.xlm

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <AnalogClock
        android:id="@+id/relojanalogico"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

</LinearLayout>
```



c. Menú

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/action_settings"
        android:checkable="true"
        android:enabled="true"
        android:numericShortcut="1"
        android:showAsAction="ifRoom"
        android:orderInCategory="100"
        android:title="Animacion"
        android:titleCondensed="Animaccc"
        android:visible="true"/>
</menu>
```

3. Creación de las clases que controlaran los xml, además de las clases de los métodos para realizar las transacciones entre pages.

a. Primera clase la PageChange: esta clase implementa a OnPageChangeListener se generan los métodos que necesita la clase, a los cuales se les realizan unos cambios

```
import android.support.v4.view.ViewPager.OnPageChangeListener;

public class PageChange implements OnPageChangeListener {
	int currentIndex = 0;
	@Override
	public void onPageScrollStateChanged(int arg0) {
		// TODO Auto-generated method stub
	}
	@Override
	public void onPageScrolled(int arg0, float arg1, int
arg2) {
		// TODO Auto-generated method stub
	}
	@Override
	public void onPageSelected(int arg0) {
		currentIndex = arg0;
	}
	//agregar
	public int getCurrentIndex() {
		return currentIndex;
	}
```

Código en amarillo es el que se debe de agregar.

b. Segunda clase la PageAdapter esta clase implementa a FragmentPagerAdapter se generan automáticamente los métodos que necesita la clase, a los cuales se les realizan unos cambios.

```
import java.util.ArrayList;
import java.util.List;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

public class PagerAdapter extends FragmentPagerAdapter {
	List<Fragment> fragments = null;
	public PagerAdapter(FragmentManager fm) {
		super(fm);
		fragments = new ArrayList<Fragment>();
		// TODO Auto-generated constructor stub
	}
	public void addFragment(Fragment fragment){
		fragments.add(fragment);
	}
	@Override
	public Fragment getItem(int arg0) {
		// TODO Auto-generated method stub
		return fragments.get(arg0);
	}
	@Override
	public int getCount() {
		// TODO Auto-generated method stub
		return fragments.size();
	}
}
```

Código en amarillo es el que se debe de agregar.

c. Tercera clase la PageTransformer esta clase implementa a ViewPager.PageTransformer se genera un método y se agregan las funciones a la clase.

El método transformPage ofrece la oportunidad de que la aplicación aplique una transformación personalizada a las páginas vistas utilizando propiedades de animación.

```
import android.view.View;
import android.support.v4.view.ViewPager;

public class PageTransformer implements ViewPager.PageTransformer
{

  private static float MIN_SCALE = 0.25f;
  private static float MIN_ALPHA = 0.25f;

  public void transformPage(View view, float position) {
      int pageWidth = view.getWidth();
      int pageHeight = view.getHeight();
      if (position < -1) {
          view.setAlpha(0.0f);
      } else if (position <= 1) {
          float scaleFactor =
                   Math.max(MIN_SCALE, 1 - Math.abs(position));
          float vertMargin = pageHeight * (1 - scaleFactor) / 2;
          float horzMargin = pageWidth * (1 - scaleFactor) / 2;
          if (position < 0) {
              view.setTranslationX(horzMargin - vertMargin / 2);
          } else {
              view.setTranslationX(-horzMargin + vertMargin / 2);
          }
          view.setScaleX(scaleFactor);
          view.setScaleY(scaleFactor);
          view.setAlpha(MIN_ALPHA +
                  (scaleFactor - MIN_SCALE) /
                  (1 - MIN_SCALE) * (1 - MIN_ALPHA));

      } else {
          view.setAlpha(0.0f);
      }
```

Recibe la página y la posición.

Con esta clase logramos dar el efecto de carrusel que tienen páginas al cambiar una a otra

Si se quisiera no se agrega

d. Cuarta clase la PlainColor esta clase que extiende de **Fragment**.

```
import android.graphics.Color;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.AnalogClock;

public class PlainColor extends Fragment {
      private static final String PLAIN_COLOR_FRAG_ARG_COLOR = "color";
private static final String PLAIN_COLOR_FRAG_ARG_ALPHA = "alpha";

int color = Color.GREEN;
View view = null;
AnalogClock clock = null;
Bundle viewRecreateState = new Bundle();

public static PlainColor newInstance(int color) {
      PlainColor frag = new PlainColor();
  Bundle frag1Args = new Bundle();
  frag1Args.putInt(PLAIN_COLOR_FRAG_ARG_COLOR, color);
  frag.setArguments(frag1Args);
  frag.setRetainInstance(true);
  return frag;
}
@Override
public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  this.color        =        (getArguments()        !=        null)        ?
getArguments().getInt(PLAIN_COLOR_FRAG_ARG_COLOR) : Color.GRAY;
}
@Override
public  View  onCreateView(LayoutInflater  inflater,  ViewGroup  container,
Bundle savedInstanceState) {
  view = inflater.inflate(R.layout.reloj, container, false);
  clock = (AnalogClock) view.findViewById(R.id.relojanalogico);
  clock.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
      vanishClock();
    }
  });
  if (savedInstanceState != null)
    viewRecreateState.putAll(savedInstanceState);
  restoreClockState();
  view.setBackgroundColor(color);
  return view;
}
```

Inicialización de las variables.

Este método hace que al darle click en el reloj se desvanezca.

```
@Override
public void onDestroyView() {
  viewRecreateState.putFloat(PLAIN_COLOR_FRAG_ARG_ALPHA,
clock.getAlpha());
  super.onDestroyView();
}
private void restoreClockState() {
  float alpha = 1.0f;
  if (viewRecreateState != null &&
      viewRecreateState.containsKey(PLAIN_COLOR_FRAG_ARG_ALPHA))
    alpha = viewRecreateState.getFloat(PLAIN_COLOR_FRAG_ARG_ALPHA);
  clock.setAlpha(alpha);
}
public void startAnimationOnClock() {
  Animation anim = AnimationUtils.loadAnimation(getActivity(),
      android.R.anim.fade_out);
  clock.startAnimation(anim);
}
public void vanishClock() {
  if (clock.getAlpha() < 0.4f)
    clock.setAlpha(1.0f);
  else
    clock.setAlpha(clock.getAlpha() - .2f);
}
@Override
public void onSaveInstanceState(Bundle outState) {
  outState.putFloat(PLAIN_COLOR_FRAG_ARG_ALPHA, clock.getAlpha());
  super.onSaveInstanceState(outState);
}
}
```

Limpia los recursos asociados a su vista.

e. MainActivity es la activity principal, la que ejecuta nuestra aplicación, la cual hará un extends de FragmentActivity, encargándose de mostrar los Fragments. Esta clase va a necesitar 3 campos, un objeto de la clase ViewPager, otro de Adapter y una instancia de PageChange. Estos 3 objetos se inicializarán en el método onCreate, y después se establecerá el adapter del ViewPager.

```
import android.os.Bundle;
import android.graphics.Color;
import android.support.v4.app.FragmentActivity;
import android.support.v4.view.ViewPager;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;

public class MainActivity extends FragmentActivity {

              ViewPager pager = null;
              PageChange pageChangeListener = null;
            @Override
            public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                pager = (ViewPager) findViewById(R.id.pager)
                PagerAdapter adapter =
               new PagerAdapter(getSupportFragmentManager());
             adapter.addFragment(PlainColor.newInstance(Color.RED));
             adapter.addFragment(PlainColor.newInstance(Color.GREEN));
             adapter.addFragment(PlainColor.newInstance(Color.BLUE));
                pageChangeListener = new PageChange();
                pager.setOnPageChangeListener(pageChangeListener);
                pager.setPageTransformer(true, new PageTransformer());
                pager.setAdapter(adapter);
            }
          @Override
              public boolean onCreateOptionsMenu(Menu menu) {
                    MenuInflater inflater = new MenuInflater(this);
                    inflater.inflate(R.menu.main, menu);
                    return super.onCreateOptionsMenu(menu);
              }
          @Override
              public boolean onOptionsItemSelected(MenuItem item) {
                    switch(item.getItemId()){
                          case R.id.action_settings:
                                startAnimationOnCurrentPage();
                          default:
                                return
super.onOptionsItemSelected(item);
                    }
              }
        private void startAnimationOnCurrentPage(){
            PagerAdapter adapter =
                (PagerAdapter) pager.getAdapter();
            PlainColor fragment =
                (PlainColor)
adapter.getItem(pageChangeListener.getCurrentIndex());
            fragment.startAnimationOnClock();
          }

        }
```

Colores para los view

Cambio de view

Llamada a la animación

Crear la opción del menú

Acción a la opción del menú seleccionada

Método, lo que realiza es volver a cargar el fragment del reloj