

MANUAL DE PROGRAMADOR DEL JUEGO ZOMBI



ANDROID

Hoy en día millones de dispositivos móviles en más de 190 países en todo el mundo, de todos esos millones de dispositivos móviles realizan descargar de aplicaciones para celulares móviles desde google play, que a muchos desarrolladores a causas de esas descargas les está impulsando a desarrollar todo tipo de aplicación móvil.

Por eso yo apporto mi propia experiencia para poder aprender más sobre lo que es desarrollar aplicaciones móviles y desarrollar más apps, en este manual de programador les enseñare todos los pasos de mi juego **ZOMBI**.

PRIMER PASO: EMPEZANDO

Para poder programar este juego debes tener el Jdk de java, eclipse, SDK de Android, y descargar el plugin de Android y por ultimo un emulador de Android. Y lo podrás descargar de los siguientes links

Ide java: <http://www.oracle.com/technetwork/es/java/javase/downloads/jdk7-downloads-1880260.html/>

Eclipse: <https://eclipse.org/downloads/>

Sdk Android: <https://developer.android.com/sdk/installing/index.html>

Plugin de Android: <http://developer.android.com/sdk/installing/installing-adt.html>

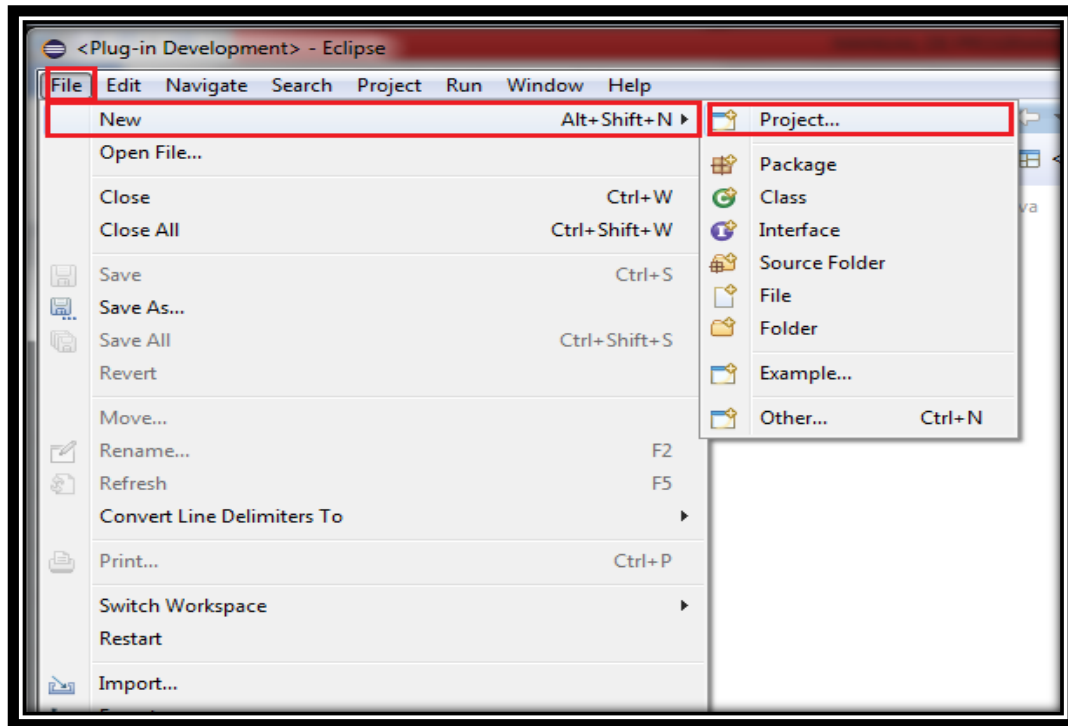
Y para que aprenda estos instaladores les dejare un video:

Video: <https://www.youtube.com/watch?v=loFg-jRvx6A>

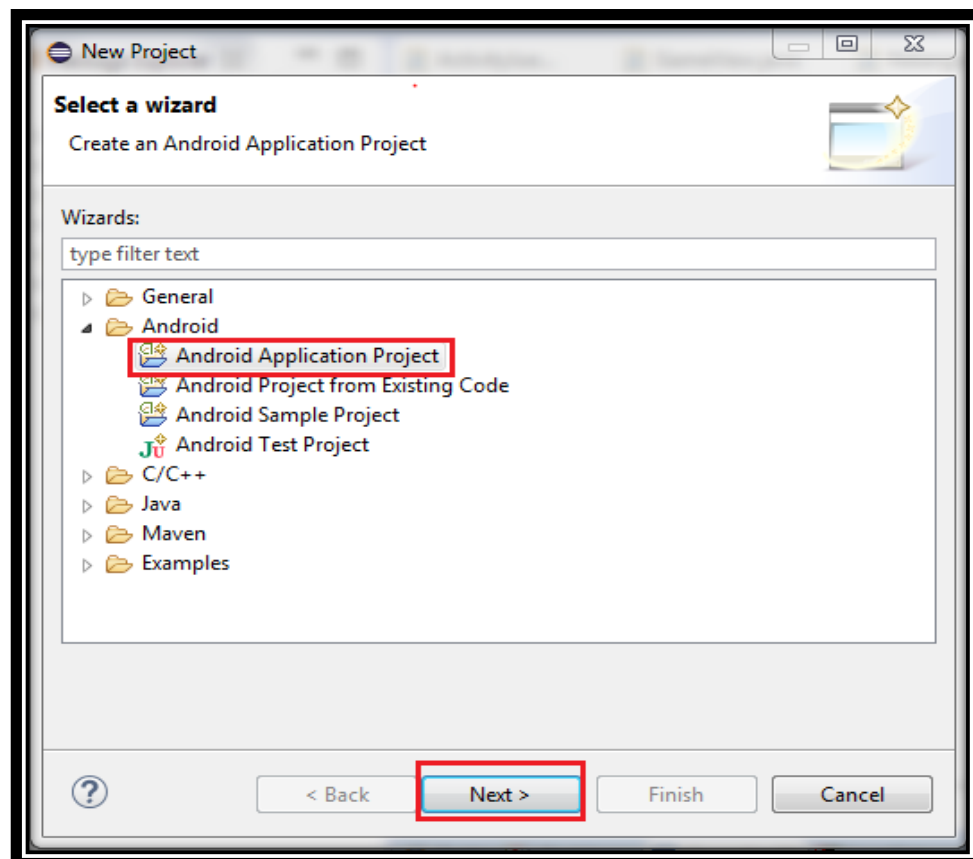
Ya teniendo instalado los programas mencionados veremos cómo funciona eclipse.

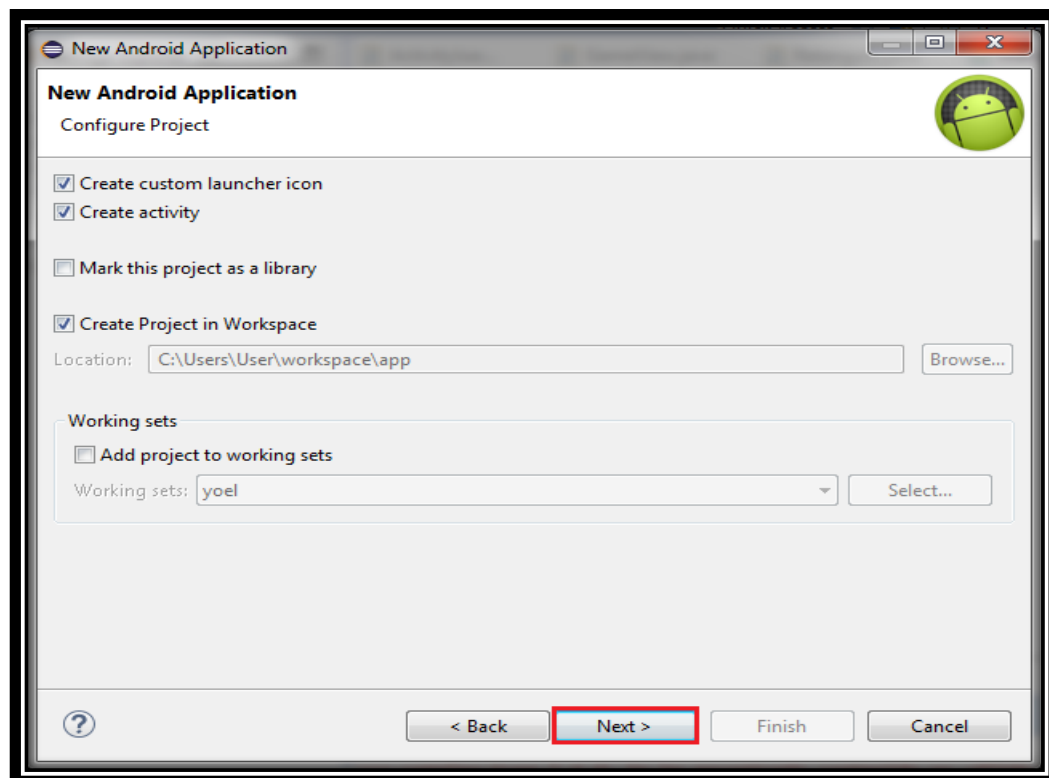
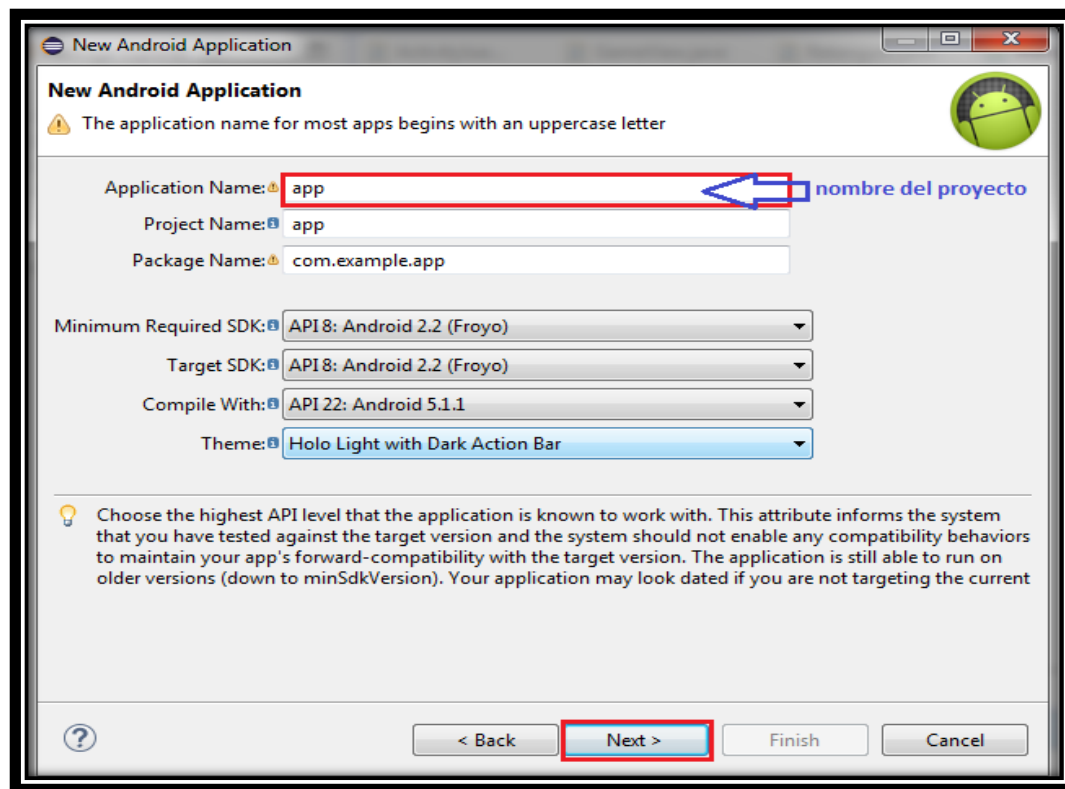
SEGUNDO PASO: CREAR UN PROYECTO EN ECLIPSE PARA CON ANDROID

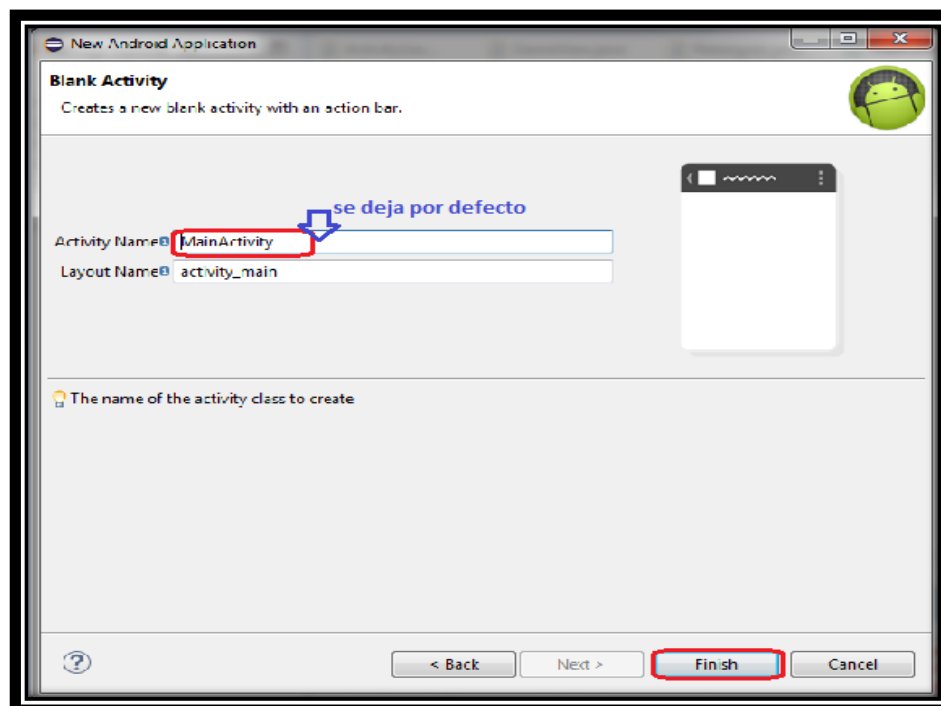
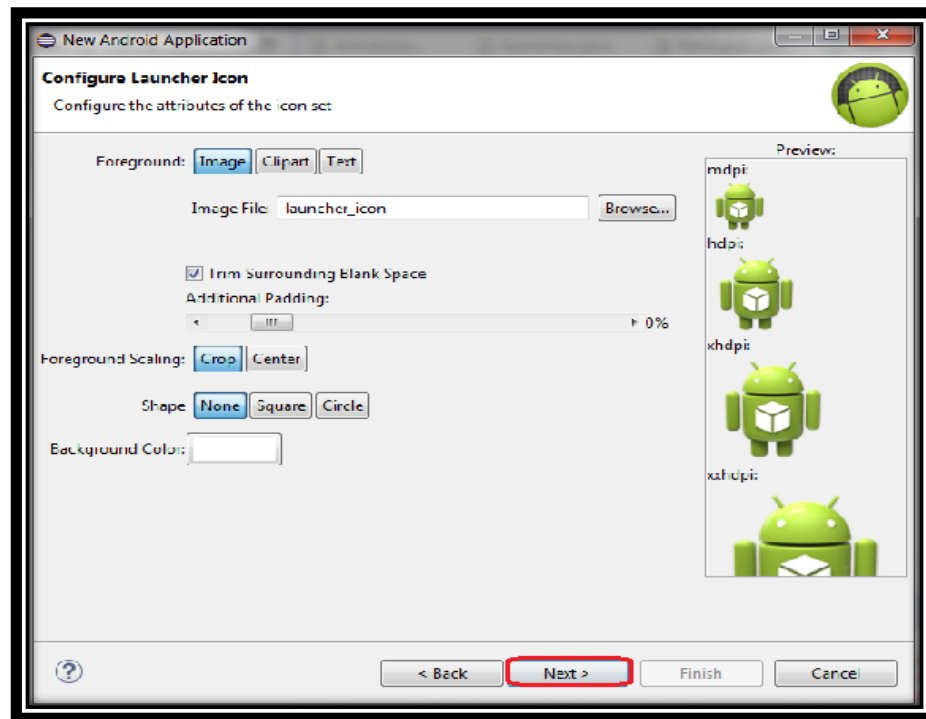
Hacer los siguientes pasos

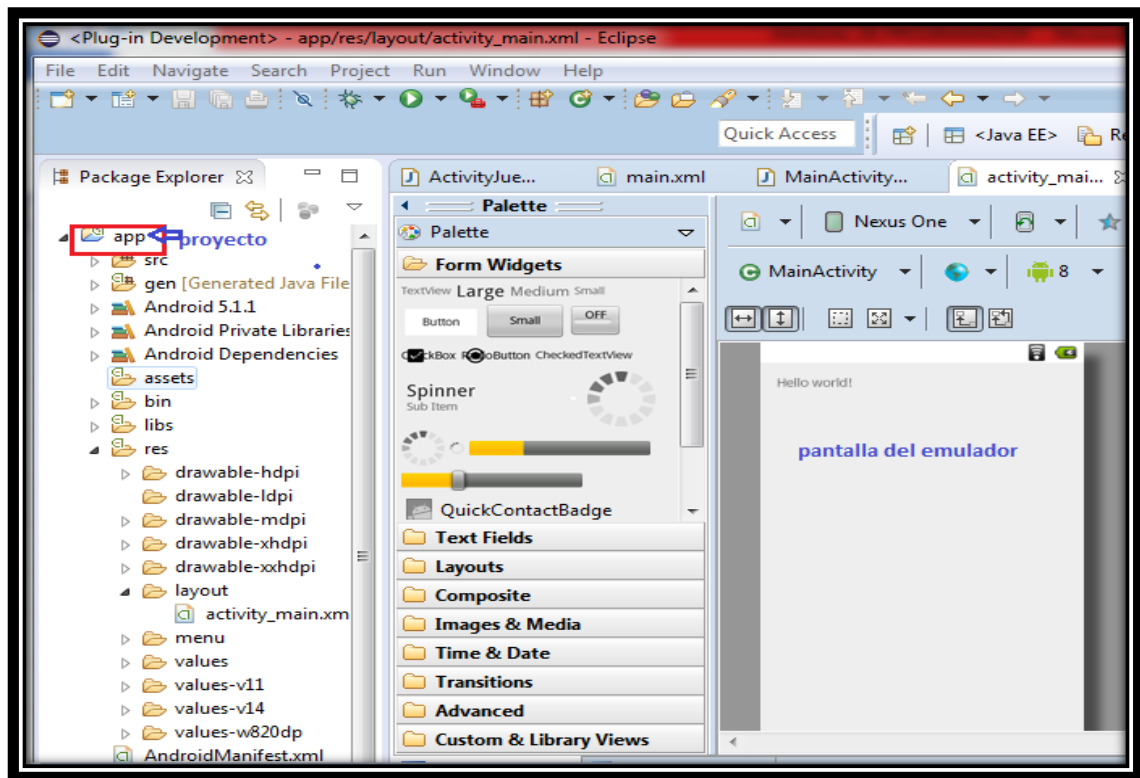


Después hacer el siguiente paso





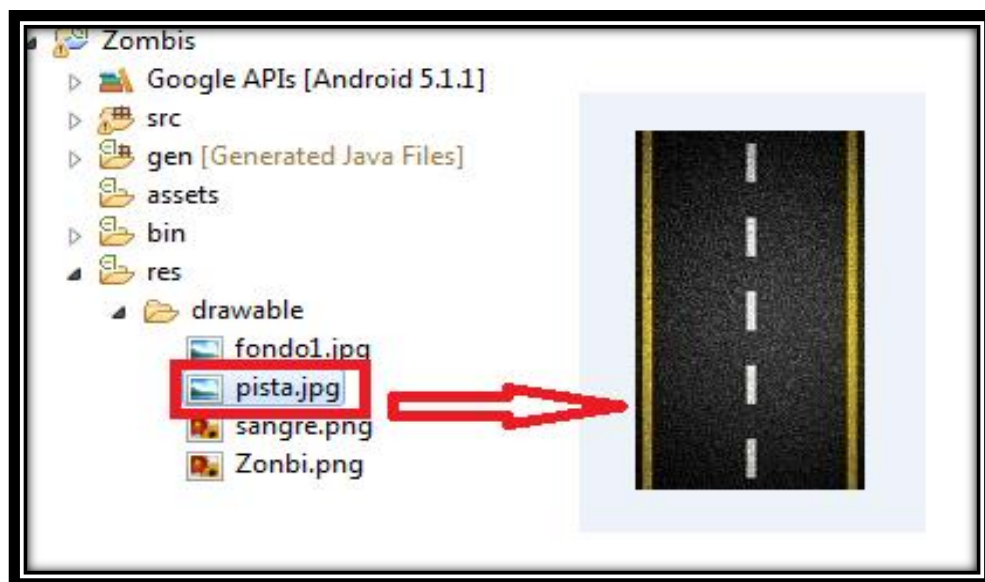




Las capturas anteriores es un ejemplo de cómo se crea un proyecto, en el tercer paso are de mi proyecto que ya estaba creado anteriormente (**ZOMBI**)

TECER PASO: PONER EL FONDO DE PANTALLA

Ponemos la imagen en la siguiente carpeta del proyecto **ZOMBI**



En seguida crearemos una clase (**GameView.java**) y un método para cargar imágenes



El siguiente es main principal **ActivityJuego** que va a instanciar la clase **GameView**



Declarar variables y librerías en clase **GameView**

```

1 package com.zombi;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import unifran.ely.R;
7 import android.content.Context;
8 import android.graphics.Bitmap;
9 import android.graphics.BitmapFactory;
10 import android.graphics.Canvas;
11 import android.graphics.Color;
12 import android.graphics.Paint;
13 import android.media.MediaPlayer;
14 import android.util.Log;
15 import android.view.MotionEvent;
16 import android.view.View;
17 // clase principal
18 public class GameView extends View implements Runnable {
19     private final static int INTERVAL = 45;
20     private boolean running = true;
21     private Paint paint;
22     private Zombi[] zombi;
23     private int puntos = 0;
24     private long tiempo = System.currentTimeMillis();
25     private List<TempSprite> temps = new ArrayList<TempSprite>();
26     private boolean iniciarJuego = false;
27     private Bitmap bmpFondo, sangre;
28     private static Bitmap bmp;

```

importar librerías
necesarios que se van
a necesitar en toda la
clase **GameView**

declaracion de variables

Método para insertar sonido en la clase **GmeView**

```

//metodo para insertar sonido
this.sonido = MediaPlayer.create(context, R.raw.pong);

```


Método para iniciar el juego en la clase **GameView**

```
//metodo para iniciar juego
public void iniciaJuego() {
    zombi = new Zombi[getHeight()/50]; //la posicion de los zombis en 50% de pantalla
    for (int i = 0; i < zombi.length; i++) {
        int y = i*-50;
        int x = (int) (Math.random()*(getWidth()-bmp.getHeight()));
        zombi[i] = new Zombi(x, y, bmp.getWidth()/2, bmp.getHeight()/3, bmp);
    }

    // metodo para insertar psita a la pantalla
    bmpFondo = BitmapFactory.decodeResource(getResources(), R.drawable.pista);
    bmpFondo = Bitmap.createScaledBitmap(bmpFondo, getWidth(), getHeight(), true);

    iniciarJuego = true;
}
```

Método para dibujar imágenes, color y texto en la clase **GmeView**

```
//invoca el método draw
postInvalidate();
}
// metodo para dibujar el fondo, color, y el texto
public void draw(Canvas canvas) {
    super.draw(canvas);
    if (iniciarJuego==false) {
        iniciaJuego();
    }

    //escoje el color de fondo
    //canvas.drawColor(Color.GREEN);
    canvas.drawBitmap(bmpFondo, 0, 0, paint);

    for (int i = temps.size() - 1; i >= 0; i--) {
        temps.get(i).onDraw(canvas);
    }
    //define el color escogido
    paint.setColor(Color.RED);
    for (int i = 0; i < zombi.length; i++) {
        zombi[i].draw(canvas, paint);
    }
}
```