

Este código nos permite **ejecutar cualquier página web como una app nativa en Android**, en el código se incluye la configuración para que funcione **el botón “atrás” de los teléfonos android**, así como **eliminar la barra de títulos** y queda listo para abrir todos los **enlaces de una página web en la APP y los externos en el navegador** (esto es muy importante para abrir enlaces de redes sociales, etc, fuera de la app como debe de ser).

El tutorial (en video) lo puedes encontrar en el siguiente enlace: <http://youtu.be/kzLmjGBFkcY>

También te recomiendo ver los siguientes videos:

- Android Studio - Descargar e instalar - <http://youtu.be/CwPCwwydE68>
- Android Studio - Crear emulador y abrir mi primer App (Hola Mundo!) - <http://youtu.be/hlQrllf0E6E>
- Android Studio - Configurar emulador en tiempo real 100% rapido (Genymotion) - <http://youtu.be/zc8-PmVhjzw>

Estamos editando el video en donde se muestra cómo exportar el APK

Suscribete a mi canal para recibir las actualizaciones de los videos: <http://goo.gl/EAjiPb>

PASO 1

\app\src\main\AndroidManifest.xml

```
<!-- INI AGREGADO -->
<!-- Otorgamos permisos de Internet a nuestra App -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- FIN AGREGADO -->
```

PASO 2

\app\src\main\res\layout\activity_main.xml

```
<!-- INI AGREGADO -->
<!-- Creamos una actividad webview para mostrar nuestra pagina web -->
<WebView
    android:id="@+id/activity_main_webview"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
<!-- FIN AGREGADO -->
```

PASO 2.1

Entrar en modo Design y borrar el padding

PASO 3

\app\MainActivity.java

```
// INI AGREGADO
private WebView mWebView;
// FIN AGREGADO
```

PASO 3.1

```
// INI AGREGADO
mWebView = (WebView) findViewById(R.id.activity_main_webview);

// Activamos Javascript
WebSettings webSettings = mWebView.getSettings();
webSettings.setJavaScriptEnabled(true);

// Url que carga la app (webview)
mWebView.loadUrl("http://google.com");

// Forzamos el webview para que abra los enlaces internos dentro de la la APP
mWebView.setWebViewClient(new WebViewClient());

// Forzamos el webview para que abra los enlaces externos en el navegador
mWebView.setWebViewClient(new MyAppWebViewClient());
// FIN AGREGADO
```

PASO 3.2

```
// INI AGREGADO
@Override
// Detectar cuando se presiona el botón de retroceso
public void onBackPressed() {
    if(mWebView.canGoBack()) {
        mWebView.goBack();
    } else {
        super.onBackPressed();
    }
}
// FIN AGREGADO
```

PASO 4

\\app\\MyAppWebViewClient.java

Creamos nueva clase con el nombre "MyAppWebViewClient" y OK.
Borramos todo el codigo de la clase y ponemos el siguiente:

```
// INI AGREGADO
public class MyAppWebViewClient extends WebViewClient {

    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {

        // Url base de la APP (al salir de esta url, abre el navegador) poner como se muestra, sin http://
        if(Uri.parse(url).getHost().endsWith("google.com")) {
            return false;
        }

        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
        view.getContext().startActivity(intent);
        return true;
    }
}
// FIN AGREGADO
```

PASO 5

\\app\\src\\main\\res\\values\\styles.xml

```
<!-- INI AGREGADO -->
<!-- Ocultamos la barra de titulo -->
<item name="android:windowNoTitle">true</item>
<!-- FIN AGREGADO -->
```