

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

AdictosAlTrabajo

Temporada Completa de Terrakas
terrakas.com



autentia
Soporte a desarrollo informático
Hosting patrocinado por **enREDados**

Entra en Adictos a través de

E-mail:

Contraseña:

[Deseo registrarme](#)
[Olvídate mi contraseña](#)

- [Inicio](#)
- [Quiénes somos](#)
- [Formación](#)
- [Comparador de salarios](#)
- [Nuestros libros](#)
- [Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) [Android Flavors](#)



Francisco J. Arroyo

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en **Autentia**: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

[Ver todos los tutoriales del autor](#)



Fecha de publicación del tutorial: 2014-04-21

Tutorial visitado 1 veces [Descargar en PDF](#)

Android Flavors

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Manos a la obra.
 - 3.1 Recursos.
 - 3.2 Código.
 - 3.3 Configuración.
 - 3.4 Ejemplo.

1. Introducción

Con el desarrollo de **Android Studio** ha venido un cambio a nivel interno que ha añadido más flexibilidad al desarrollo de aplicaciones para **Android**. Este cambio ha sido el remplazo de **Ant** por **Gradle**. Al realizar el cambio, la gente de Android ha tenido (y están) que desarrollar un plugin para Gradle para que a partir del código fuente que realicemos, se puedan construir aplicaciones para Android. Y con esto, una de las funcionalidades de las que han dotado al plugin de Gradle es el uso de **Flavors** (sabores).

¿Y para que sirve esto? Seguro que habéis visto en más de una ocasión cuando habéis querido instalar alguna app desde Google Play (o cualquier otro sitio) que hay una versión "lite" y una "pro", o una versión con publicidad y otra con pagos "in app". Eso, en la mayoría de las ocasiones implicaba tener dos proyectos, extraer el máximo código posible a librerías para compartirlas, en otras ocasiones implicaba crear tareas de ant o perfiles de maven para realizar ese trabajo. Gracias a esta funcionalidad, se simplifica mucho nuestro trabajo.

En este tutorial vamos a ver como podemos hacer uso de esta funcionalidad, fácil fácil :P

2. Entorno

- Hardware
 - MacBook Pro
 - Intel Core i7 2Ghz
 - 8GB RAM
 - 250GB SSD
 - Sistema Operativo: Mac OS X (10.9)
- Software
 - Android Studio 0.5.4
 - Gradle 1.10

3. Manos a la obra.

Nuestro objetivo va a ser crear una aplicación muy simple que haga uso de los flavors. La aplicación consistirá de una imagen y un texto que serán distintos dependiendo del flavor que construyamos.

El primer paso es crear una aplicación con Android Studio, basta con que dejéis los valores por defecto.

Una vez que tenemos el proyecto creado, vamos a añadir los dos sabores o flavors.

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» [Aprende gratis ReactiveCocoa](#)

» [Checklist de Scrum de Autentia](#)

» [Autentia estrena web y celebra el X Cycling Day](#)

» [Buscamos personal para Autentia y nuestros clientes \(10-Marzo-2014\)](#)

» [Charla de Auto Layout en nuestra oficina](#)

[Histórico de noticias](#)

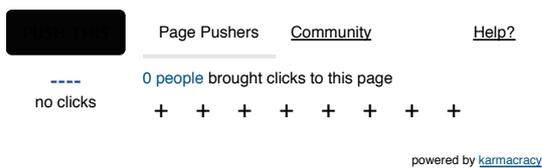
Últimos Tutoriales

» [Maven, Jenkins, Sonar y tests de integracion](#)

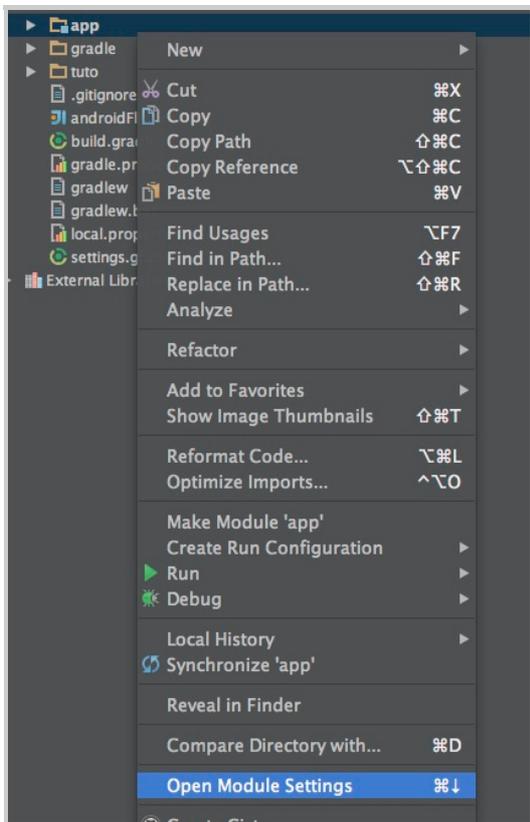
» [Spring Ldap: gestión de transacciones y configuración de un pool de conexiones](#)

» [Spring Ldap: operaciones básicas haciendo uso del soporte de plantillas](#)

» [Crear un plugin para Android en PhoneGap](#)



Botón derecho encima del módulo de la aplicación y clickamos en "Open Module Settings".



» Implementando tu propio Writable en Hadoop

Últimos Tutoriales del Autor

» Cómo añadir Volley (librería de Android) en Android Studio

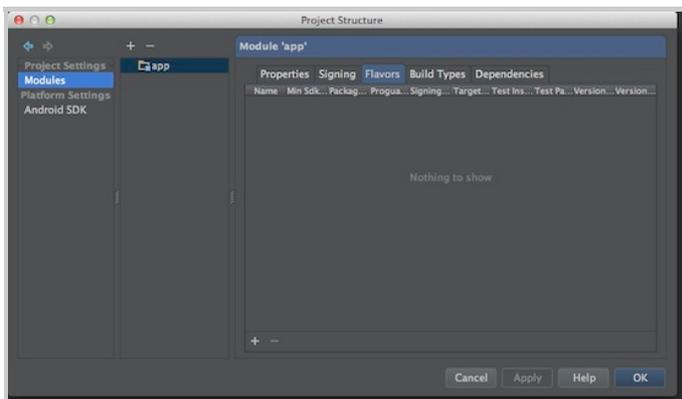
» Integrar Barcode Scanner en nuestra aplicación Android

» Jugando con JSON en Java y la librería GSON. Parte 2

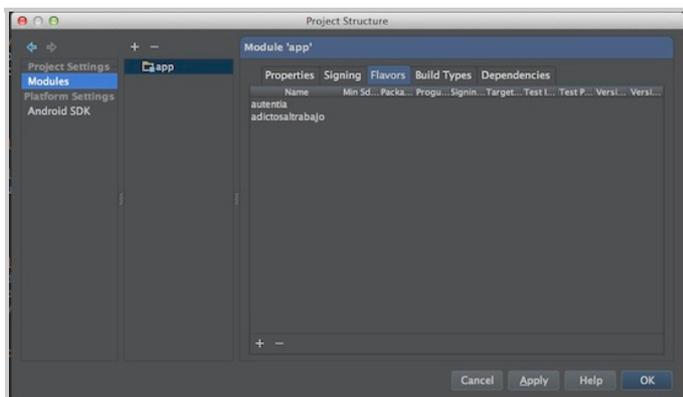
» Android Beam

» Como hacer nuestros test más legibles con Hamcrest

A continuación vamos a Flavors



y por último añadimos los dos flavors que vamos a usar: autentia y adictosaltrabajo

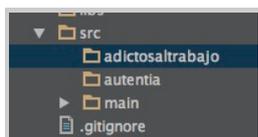


Aceptamos y tendremos preparado el proyecto para usar los dos flavors. Un vistazo a build.gradle nos lo confirma

```

view plain print ?
01. //Continua
02. }
03.     productFlavors {
04.         autentia {
05.             }
06.         adictosaltrabajo {
07.             }
08.     }
09. //Continua
    
```

El siguiente paso es crear las carpetas en donde alojaremos el código dependiente de cada flavor. Las carpetas se han de colocar bajo la carpeta **src** y deben tener el mismo nombre que hemos usado al crear los flavors. El código común se seguirá ubicando bajo **main**

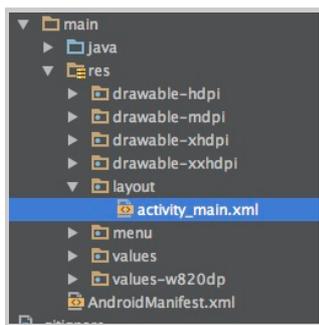


Cuando construyamos la aplicación, el plugin de Android escogerá los fuentes de la carpeta del flavor que esté construyendo y los mezclará con el directorio main dando lugar a la aplicación resultante.

3.1 Recursos

Como hemos dicho al principio, vamos a crear una aplicación que muestre una imagen dependiendo del flavor. Para ello vamos a usar un layout común a los dos flavors que contendrá un **ImageView** en el que mostraremos la imagen.

Modificamos el layout **activity_main** para cambiar el texto por una imagen. Fijaos que estoy tocando el layout que cuelga de main (y no en los flavors).



```

view plain print ?
01. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
02.                 xmlns:tools="http://schemas.android.com/tools"
03.                 android:layout_width="match_parent"
04.                 android:layout_height="match_parent"
05.                 android:paddingLeft="@dimen/activity_horizontal_margin"
06.                 android:paddingRight="@dimen/activity_horizontal_margin"
07.                 android:paddingTop="@dimen/activity_vertical_margin"
08.                 android:paddingBottom="@dimen/activity_vertical_margin"
09.                 tools:context="com.annamolly.raspberrypi.commander.android.androidflavors.app.MainAc
10.
11.     <ImageView
12.         android:layout_width="wrap_content"
13.         android:layout_height="wrap_content"
14.         android:id="@+id/ivFlavor"
15.         android:layout_centerVertical="true"
16.         android:layout_centerHorizontal="true"
17.         android:src="@drawable/logo"/>
18. </RelativeLayout>
    
```

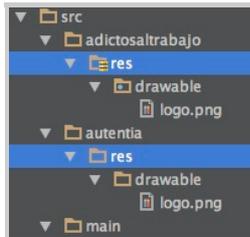
Lo que vamos a hacer a continuación es añadir los correspondientes logos a los dos flavors que hemos creado.

Aunque estemos usando otro directorio para dejar los fuentes de la aplicación, tenemos que usar la misma estructura que si estemos usando main. Fijaros en el detalle de la carpeta **res** en la que Android Studio ya me la marca con un icono de recursos. Es posible que Android Studio os marque en rojo indicando que no existe el logo, ignorarlo, es un bug.

En mi caso ha quedado así:



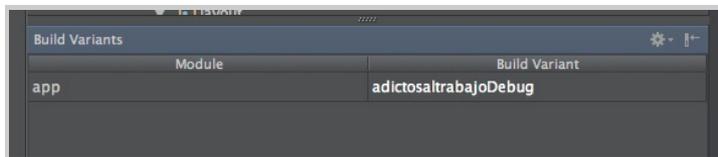
Si os fijáis, AS me ha marcado sólo una de las dos carpetas **res** como recursos. Eso es porque tengo seleccionado en Android Studio ese flavor



Y si la siguiente pregunta es: ¿y como cambio de flavor?. Si buscáis por el borde de AS encontraréis un botón que pone **Build Variants** en donde podemos establecer el activo. Build Variants es un concepto que se compone de Build Types y Flavors, pero en este tutorial solo abarcaremos flavors,



y en mi caso tengo activado la variante debug del flavor adictosaltrabajo



Aquí podemos hacer dos cosas: Darle al botón de play y ejecutar en el emulador (o en un dispositivo) la versión del flavor que tenemos seleccionado o también podemos ejecutar gradlew build y generar un apk por cada flavor.



3.2 Código

No solo podemos configurar recursos, también podemos tener código dependiente de cada flavor. Vamos a crear una clase que nos va a devolver un texto diferente dependiendo del flavor.

La implementación para el flavor adictosaltrabajo es:

```

view plain print ?
01. package com.adictosaltrabajo.tutoriales.android.flavors;
02.
03. public class PhraseGenerator {
04.
05.     public static String get() {
06.         return "Visítanos en http://www.adictosaltrabajo.com/";
07.     }
08. }
    
```

y la implementación para el flavor autentia es:

```

view plain print ?
01. package com.adictosaltrabajo.tutoriales.android.flavors;
    
```

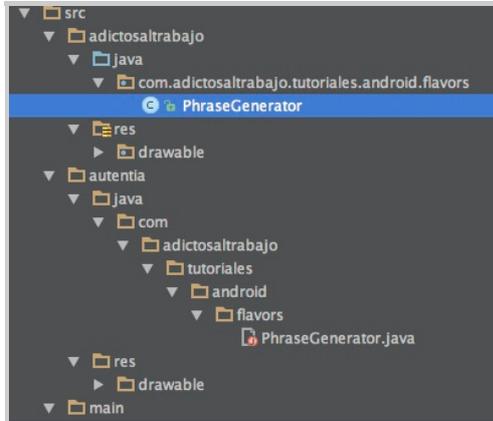
```

02.
03. public class PhraseGenerator {
04.
05.     public static String get() {
06.         return "Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos";
07.     }
08. }

```

Es un ejemplo muy sencillo en el que espero que se vea que podemos tener distintas implementaciones en los distintos flavors.

Mi estructura de directorios queda así:



Ocurre como con el directorio **res**: Como tengo seleccionado el flavor de **adictosaltrabajo**, AS interpreta que es un paquete y me lo pinta así y en el caso de **autentia** me lo muestra como carpetas.

Modificamos de nuevo nuestro layout para añadir un textview

```

view plain print ?
01. ....
02.
03.     <TextView
04.         android:layout_width="wrap_content"
05.         android:layout_height="wrap_content"
06.         android:textAppearance="?android:attr/textAppearanceMedium"
07.         android:text="Medium Text"
08.         android:id="@+id/tvText"
09.         android:layout_alignParentBottom="true"
10.         android:layout_centerHorizontal="true" />
11.
12. </RelativeLayout>

```

y modificamos la clase MainActivity para establecer el valor del textview

```

view plain print ?
01. @Override
02. protected void onCreate(Bundle savedInstanceState) {
03.     super.onCreate(savedInstanceState);
04.     setContentView(R.layout.activity_main);
05.     //modificamos el texto del textview
06.     ((TextView)findViewById(R.id.tvText)).setText(PhraseGenerator.get());
07. }

```

3.3 Configuración

También podemos modificar valores de la configuración dependiendo del flavor.

Al principio del tutorial, veíamos como al añadir los flavors, AS nos modificaba el fichero **build.gradle**. Esto nos abre otra posibilidad de personalización.

Os pongo un par de ejemplos, pero también se podría cambiar el nombre del paquete, certificados de firma..

```

view plain print ?
01. ....
02.     productFlavors {
03.         autentia {
04.             //Numero de version 20
05.             versionCode 20
06.         }
07.         adictosaltrabajo {
08.             versionCode 18
09.         }
10.     }
11. }
12.
13. dependencies {
14.     //Dependencia sólo para el flavor autentia
15.     autentiaCompile 'commons-lang:commons-lang:2.6'
16. }

```

3.4 Ejemplo

Y así es como queda cada una de los sabores de nuestra aplicación.

Autentia



Adictos al trabajo



4. Conclusiones

Y hasta aquí hemos llegado con **flavors**. Si por alguna razón necesitamos generar distintas versiones de la aplicación, vemos que solo necesitamos un mínimo de configuración y seguir una estructura de directorios, el resto nos lo gestiona el sdk de Android. Sencillo ¿no? ;P

Para cualquier comentario, duda o sugerencia, tenéis el formulario que aparece a continuación.

Un saludo.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share |

 0

¡Anímate y coméntanos lo que pienses sobre este **TUTORIAL**!

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2014 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

