

# Convenciones De Nombres En Android

Escrito por James Revelo

¡Bienvenido al minibook de convenciones de nombres en Android!

En esta serie de lineamientos verás una recopilación de reglas para nombrar los archivos Java y de recursos en tus proyecto Android Studio,

Con ello nos referimos a los estilos para nombrar archivos de layouts, iconos, drawables, etc. O incluso como escribir nombres de clases, métodos, campos y demás.

## 1. Nombrar Archivos

En esta guía usaremos el idioma Inglés para generar los nombres de los archivos, ya que la mayoría de veces es más cómodo que el español en usos de caracteres.

Es compacto al describir una sustantivo o acción y no usa marcas de acento como las tildes.

(No obstante, si tu lo deseas, puedes crear reglas de estilo basados en español que te sean de utilidad)

### 1.1 Nombrar Clases Java

Recuerda que el mismo nombre de las clases Java debe ser el mismo de su archivo.

Para nombrarlas usa la notación *UpperCamelCase*. Donde cada palabra debe llevar su primera letra en mayúscula.

#### Ejemplos

- LoginPresenter.java
- SalesmenActivity.java
- OrderItem.java
- ProductsViewModel.java

Facilita el entendimiento del archivo, añadiendo a la clase, el nombre del componente asociado.

### Ejemplos

- PetsActivity.java
- TrackingContentProvider.java
- SyncService.java
- GetProductAsyncTask.java

## 1.2 Nombrar Recursos

Usa minúsculas y separa por guiones bajos cada palabra adicional.

### Ejemplos

- activity\_main.xml
- menu\_login.xml
- item\_lawyer.xml

### Layouts

Mueve el nombre del componente al inicio y luego pon su objetivo.

*[nombre\_componente]\_[objetivo]*

La siguiente tabla clarifica esta notación:

Componente	Ejemplo Nombre de la clase	Nombre del layout
Actividad	ContactsListActivity	activity_contacts_list.xml
Fragmento	AddToolFragment	fragment_add_tool.xml
Diálogo	ChangeGroupNameDialog	dialog_change_group_name.xml
Item de adaptador	---	item_dog.xml

Layout parcial	---	partial_user_form.xml
Menús	menu_[nombre_actividad]	menu_login.xml

## Values

Normalmente los recursos con valores elementales deben escribirse en plural asociando su naturaleza:

Recurso	Nombre de archivo
Strings	values/strings.xml
Estilos	values/styles.xml
Colores	values/colors.xml
Dimensiones	values/dimens.xml
Enteros	values/integers.xml
Booleanos	values/bools.xml

## Iconos

Por el lado de los iconos, dependiendo del lugar a donde pertenezcan, así mismo los clasificamos con prefijos.

Contexto	Prefijo	Ejemplo
Contenido	ic_	ic_phone

Iconos de aplicación	ic_launcher_	ic_launcher_supervisor
Menús y App Bar	ic_menu_	ic_menu_add
Barra de estado	ic_stat_	ic_stat_sync_progress
Pestañas	ic_tab_	ic_tab_locations
Diálogos	ic_dialog	ic_dialog_pick_date

## 2. Nombrar Código

### 2.1 Código Java

Sobre este tema hay varias fuentes. Basta que pongas en Google [“java naming conventions”](#) y verás la cantidad de estilos.

No obstante, la siguiente tabla resume el estilo más usado para cada elemento de código:

Elemento	Convención	Ejemplo
Clases	Notación <i>UpperCamelCase</i> . Claramente debe ser un sustantivo	UseCase, DataMaganer, XmlParser
Métodos	Notación <i>lowerCamelCase</i> . Exactamente como las clases, solo que la primera es minúscula.	fetchCustomer(), downloadProfilePhoto()
Campos privados no estáticos	Notación <i>lowerCamelCase</i> . Antepón la letra <i>m</i>	mCurrentPage, mUserName, mContext

Campos privados estáticos	Notación <i>lowerCamelCase</i> . Antepón la letra <i>s</i>	sVolleySingleton, sCorrectAnswer
Otros campos y variables	Notación <i>lowerCamelCase</i>	disccountPercent, acceptButton
Nombres de constantes	Todas sus letras en mayúsculas	LATENCY, SYNC_INTERVAL, PREF_USER_PROFILE

## Siglas

Por lo general las siglas vienen todas en mayúsculas como: JSON, XML, HTML, PDO, etc.

Si vas a incluir una palabra de este tipo en el nombre de tu código Java, entonces trátala como una palabra simple.

## Ejemplos

Bien	Mal
XmlHttpRequest	XMLHTTPRequest
getCustomerId	getCustomerID
class Html	class HTML
String url	String URL
long id	long ID

## Constantes Android

Recuerda que en Android hay varios componentes que usan el mecanismo de pares clave-valor. Google nos sugiere declarar la clave como una constante.

Por ejemplo, al pasar un valor entre actividades usamos "extras" en los intents de comunicación.

La siguiente es una tabla con reglas sugeridas para varios elementos comunes:

Componente	Prefijo	Ejemplo
Preferencias	PREF_	PREF_USER_PROFILE
Bundle	BUNDLE_	BUNDLE_TABLE_NAME
Argumentos de fragmentos	ARGUMENT_	ARGUMENT_CUSTOMER_ID
Extras de intents	EXTRA_	EXTRA_QUERY_SEARCH
Acciones de intents	ACTION_	ACTION_TAKE_PHOTO
Códigos de estado	STATUS_CODE_	STATUS_CODE_PRODUCT_SELECTION

## 2.2 Código XML

En este apartado verás nuestras reglas sugeridas para los nombres de los identificadores (`android:id`) de etiquetas XML.

### Identificadores

Puedes usar el nombre del componente y luego añadir su nombre particular.

Componente	Prefijo	Id de ejemplo
Button	button_	button_add_order

ImageView	image_	image_friend_avatar
TextView	text_	text_price_label
Spinner	spinner_	spinner_age_range
Menu	menu_	menu_delete_all

O usar abreviaciones de dos o tres letras para el componente como prefijo:

Componente	Prefijo	Ejemplo
Button	bt_	bt_add_order
ImageView	iv_	iv_friend_avatar
TextView	tv_	tv_price_label
Spinner	sp_	sp_age_range
Menu	mn_	mn_delete_all

## Strings

El nombre del string (atributo `android:name`) lleva antepuesto el nombre de característica a la que pertenezca y seguido su objetivo.

### Ejemplos

- `login_user_email`
- `list_post_name_filter`
- `machine_detail_temperature`

Si el string hace parte de elementos de IU más generales, entonces puedes usar el origen como prefijo:

<b>Prefijo</b>	<b>Propósito</b>
error_	Mensaje de error
msg_	Texto de información
title_	Títulos de contenedores como actividades, fragmentos, diálogos, etc.
action_	Descripción de acciones como "Eliminar", "Ajustes", etc.

# Conclusión

Esperamos que esta pequeña guía te ayude a facilitar el nombramiento de tus archivos y el código.

Permitiéndote ser más eficiente a la hora de escanear tu código y determinar su funcionalidad.

Algunos links de referencia:

- [Code Style for Contributors - Google](#)
- [Project Guidelines – Ribot](#)
- [Best practices in Android development – Futurice](#)
- [Naming Conventions - Oracle](#)

## Feedback

Envíanos tus sugerencias y correcciones sobre esta guía a través de la página de [Contacto](#) de develou.com. Tu feedback es esencial para mejorar los contenidos del website.