

# Material Design - Un lenguaje Visual para el desarrollo Ágil de Software

Luis Vivas, Horacio Muñoz Abbate, Nicolás García Martínez,  
Mauro Cambarieri, Marcelo Petroff

Laboratorio de Informática Aplicada - Universidad Nacional de Río Negro  
{lvivas, hmunoz, ngarciam, mcambarieri,, mpetroff}@unrn.edu.ar

**Resumen.** Este trabajo presenta la incidencia de Material Design en los procesos ágiles de desarrollo de software para obtener productos de alta calidad e innovadores en diseño de interfaces, en los tiempos que la metodología exige. Expone los últimos datos que revelan la importancia en el mercado de Material Design y su aceptación en la comunidad de diseñadores, desarrolladores y usuarios. También explica la variedad de framework disponibles para implementar Material Design en aplicaciones web y nativas, además de técnicas de implementación. Por último, detalla una prueba de concepto para validar la hipótesis planteada.

**Keywords:** móviles, software libre, Android, material design, metodologías ágiles, polymer

## 1 Introducción

Desarrollar software de manera exitosa y con calidad, depende de varios factores. El incremento de las respuestas ágiles, se sustenta en los métodos, en el grupo de trabajo, en las técnicas de desarrollo, en las herramientas y por supuesto, en la tecnología.

El surgimiento de nuevos paradigmas, implementados por sus respectivos lenguajes de programación, permitieron evidenciar los beneficios en costo, tiempo y calidad, generados por la correcta elección del lenguaje de programación que aplique al dominio de problema presentado.

El auge de los lenguajes de diseño para desarrollos de software web o aplicaciones nativas, implica elegir el que resulte más eficiente, más innovador, el cual permita, además de aumentar la usabilidad y generar interfaces más atractivas, incrementar la agilidad, acortando los tiempos de entrega de soluciones y, maximizando la reusabilidad de sus componentes.

Material Design, en ese contexto, parece ser innovador lenguaje de diseño.

La principal motivación de este trabajo es, ¿Cómo el uso de Material Design mejora el desarrollo de software en un marco ágil?.

También analizamos los distintos frameworks que ofrece el mercado y cuál parece adaptarse mejor.

Nuestra hipótesis parte de que Material Design en conjunto con otro framework de desarrollo aporta una mejora sustancial en la agilidad respecto al diseño

de aplicaciones web o nativas dentro del proceso ágil logrando productos con alta calidad de usabilidad y estética, en los plazos que el cliente lo requiere.

## 2 Material Design - Conceptos Generales

### 2.1 Material Design

Material Design es un lenguaje visual que sintetiza los principios clásicos de buen diseño. Definición que Google sostiene y nosotros adherimos.

Los lineamientos mencionados anteriormente, nos facilitan la creación de aplicaciones móviles o sitios web en relación al diseño, usabilidad y calidad, consiguiendo aplicaciones que se pueden adaptar a cualquier contexto como teléfonos inteligentes, tabletas, wearables, televisores, autos o notebook.

En el evento anual que Google organiza con el nombre de I/O-2015<sup>1</sup> [2] se presentaron todas las novedades que la empresa desarrolló en el año, pero se enfatizó en Material Design haciendo hincapié en las estadísticas obtenidas desde su lanzamiento; cabe recordar que Material Design se lanzó oficialmente en el I/O-2014.

Los lineamientos que Google publicó sobre Material Design tuvieron más de 2.5 Millones de consulta. Estos se convirtieron en una guía de consulta recurrente para el 60% de estos usuarios. Las secciones más relevantes fueron tipografía, iconografía, componentes visuales como CardView<sup>2</sup> y FAB<sup>3</sup>.

Del análisis de estas estadísticas de acceso se obtiene que los lineamientos definidos no sólo son consultados por diseñadores sino que también por desarrolladores de software. Polymer [3] es el sistema de google para llevar los componentes de Material Design a la Web. Estos componentes fueron consultados por más de 2.5 Millones de desarrolladores de software y en la actualidad se han creado más de 100.000 proyectos con estos elementos.

También se presentaron datos sobre Android. Actualmente se estiman más de 250.000 aplicaciones Android nativas que implementaron Material Design. Es importante destacar que Lollipop<sup>4</sup> tiene su primer versión estable en Octubre de 2014 y es la primer versión que implementa de forma nativa Material design y si analizamos la cantidad de aplicaciones nuevas desde que Lollipop fue liberada, las aplicaciones con Material Design representan un 40% del total de las aplicaciones nuevas.

### 2.2 Android Design Library

---

<sup>1</sup> I/O siglas del inglés Innovation in the Open.

<sup>2</sup> CardView componente visual con forma de tarjeta, tiene el aspecto de un de papel suspendido en el aire.

<sup>3</sup> FAB acrónimo del inglés Float Action Button (Botón de acción flotante).

<sup>4</sup> Lollipop es el nombre clave de la versión L o 5.0 de Android.

La versión L o 5.0 con el nombre clave de Lollipop fue la primera versión de Android que implementa el concepto de Material Design. Los desarrolladores pueden implementar Material Design de forma nativa para versiones 5.0 o superiores, pero para las versiones anteriores a 5.0 tienen que usar una librería llamada AppCompat que permite tener una compatibilidad para las demás versiones. Poder generar esta compatibilidad requiere un gran esfuerzo, dado que se crean distintos xml<sup>5</sup> para implementar los componentes visuales, animaciones y transiciones para las distintas versiones.

Para solucionar este problema y facilitar la implementación de Material Design en Android, Google desarrolló Android Design Library [4]. Esta librería nos permite encapsular todo los componentes y estilos de Material Design para todas las versiones de Android. Esto es fundamental, porque nos permite aplicar Material Design sin preocuparnos la versión de Android.

### 2.3 Polymer

Polymer es conjunto de herramientas que permiten crear aplicaciones web con calidad de producción de forma repetitiva. Provee un amplio conjunto de componentes que permiten construir cualquier tipo de sitio web con el concepto de Material Design. En el evento I/O 2015 se presentó la versión 1.0 [5], primera versión estable, previo a esta versión los desarrolladores utilizaban la versión 0.5, la cual tuvo mucho éxito en cantidad y calidad de implementaciones de sitios web.

Es importante destacar el amplio catálogo de componente que provee Polymer, en la figura 1 se puede ver las principales categorías.

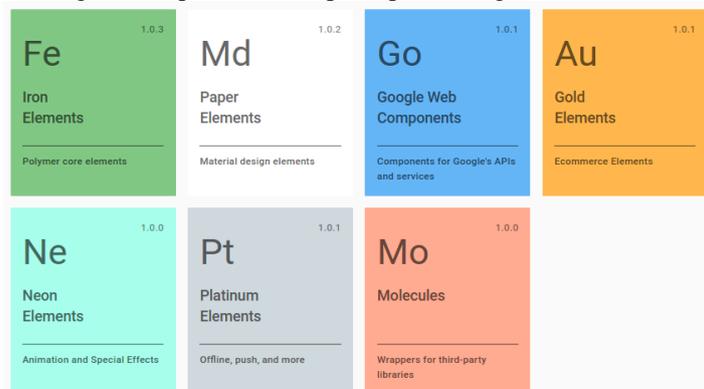


Figura 1. Catálogo de componentes de Polymer para facilitar la construcción de sitios webs con Material Design.

- **Iron Elements:** Un conjunto de elementos de utilidad visuales y no visuales. Incluye elementos para trabajar con el diseño, entrada de usuario y selección. Estos componentes se complementan con otros para proporcionar comportamiento como validaciones.

<sup>5</sup> XML, eXtensible Markup Language (lenguaje de marcas extensible) permite crear interfaces para usuarios.

- **Paper Elements:** Conjunto de elementos visuales que permiten implementar Material Design a nivel diseño. Algunos de los componentes más relevantes son `paper-fab` y `paper-material` que permite incluir el clásico botón FAB y el contenedor tipo tarjeta. Como se puede ver en la figura 2, los distintos tipos de botones que permiten crear el componente `paper-fab` dentro de un contenedor `paper-material`.

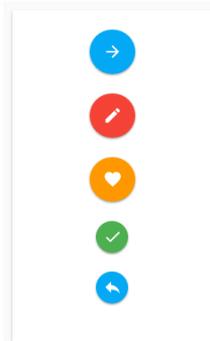


Figura 2. Botones del componente `paper-fab`

El código para generar esta vista se reduce ampliamente respecto a una implementación convencional gracias a estos componentes, el código sería:

```
<paper-material elevation="1">
  <paper-fab icon="add"></paper-fab>
  <paper-fab mini icon="favorite"></paper-fab>
  <paper-fab src="star.png"></paper-fab>
</paper-material>
```

- **Google web component:** Estos componentes facilitan la implementación de los elementos desarrollados por google como por ejemplo Google maps, hoja de cálculo, youtube, calendario entre otros. Incluir google maps en nuestro sitio es tan simple como utilizar el siguiente tag:  
`<google-map latitude="37.790" longitude="-122.390"></google-map>`
- **Gold Element:** Son los elementos para comercio electrónico. Unos de los componentes más utilizados es el de validación de tarjeta de crédito. En la figura 3, se puede observar un ejemplo con distintas tarjetas de crédito.

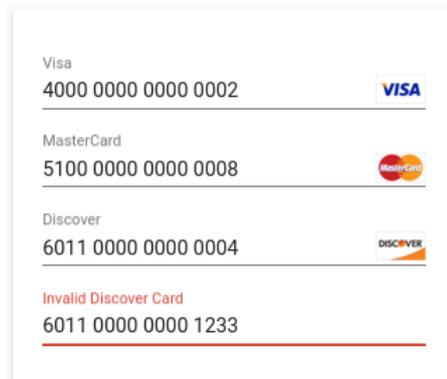


Figura 3. Ingreso para la validación de tarjetas de crédito.

El código necesario para implementar gracias a estos componentes se reduce al siguiente tag: `<gold-cc-input label="CC"></gold-cc-input>`

- **Neon Element:** Este componente brinda la posibilidad de crear animaciones. Esta implementación está basada en la especificación de la w3c de web-animations [6].
- **Platinum Element:** Estos componentes hacen posible generar aplicaciones web con funcionalidades muy similares a las aplicaciones nativas. Las funciones más destacables son la posibilidad de agregar notificaciones y la posibilidad de operar la aplicación web sin conectividad.
- **Molecules:** Son elementos que encapsulan librerías Javascripts. Este componente en la actualidad está de forma incompleta, próximamente google brindará más soporte.

## 3 Implementación de Material Design

### 3.1 Material Design para Aplicaciones Webs

Uno de los frameworks más destacados para implementar Material Design en proyectos web es sin lugar a dudas Polymer. Este proyecto está desarrollado por los mentores del concepto y la filosofía Material Design.

Cabe decir que existen otros frameworks además de Polymer que respetan todos los conceptos de Material Design y cada uno de ellos se puede adaptar mejor al equipo de trabajo disponible para llevar a cabo el proyecto web.

#### 3.1.1 Angular Material

El proyecto Angular Material [7] es una implementación de Material Design en Angular.js<sup>6</sup> [8]. Este proyecto ofrece un conjunto de componentes de interfaz de usuario reutilizables, bien probado, y accesibles basados en el sistema de Material Design. Una de las ventajas principales de este frameworks es que se puede utilizar en cualquier plataforma, ya sea PHP, JAVA o PYTHON, debido a que solo se compone de librerías de Javascripts, html y css. La comunicación entre controlador y vista es a través de objetos serializados mediante el formato json (javascript object notation, es un formato ligero para el intercambio de datos) mediante servicios de transferencia de estado representacional conocidos como servicios RESTFUL.

### 3.1.2 Bootstrap Material

Bootstrap Material es la implementación de Material Design con el framework de Bootstrap [9] [10], este framework es el más popular para el desarrollo de proyectos responsivos, característica que permite adaptarse a distintos tamaño de pantalla para dar mejor soporte a todo tipo de dispositivos.

Este frameworks es la mejor opción para equipos de desarrollo que ya tienen conocimientos con Bootstraps, además, de ser muy simple de usar. En la siguiente dirección <http://daemonite.github.io/material/> se puede ver todo un proyecto implementado con este frameworks que logra muy buena adaptación a los conceptos de Material Design.

### 3.1.3 Polymer

Polymer además de tener su especificación como se mencionó anteriormente, provee una guía de desarrollo para comenzar un proyecto base, esta guía se llama Polymer Starter Kit [11].

## 3.2. Material Design para Aplicaciones nativas Android

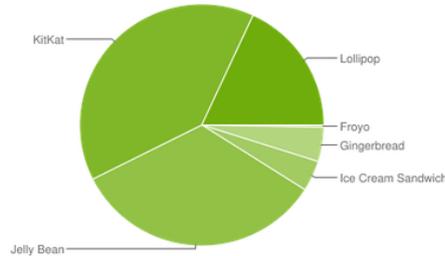
Para la implementación de Material Design en Android, como ya lo mencionamos en el apartado de conceptos, Google provee una nueva capa de librerías llamada Android Design Library que permite al desarrollador agilizar la construcción de una aplicación logrando una amplia compatibilidad de versión de android.

Unos de los problemas que enfrenta un desarrollo en Android, es la fragmentación de versiones, puede verse en la siguiente referencia [12] la cual posee las últimas estadísticas a nivel mundial. En la figura 4, se observa que android L y superiores solo tienen una 18% de la cuota del total de dispositivos Android.

---

<sup>6</sup> AngularJs es un framework que facilita el desarrollo de aplicaciones web de una sola página (SPA: Single Page Application) sobre tecnologías de navegador (JS, CSS, HTML) a través de una implementación de la arquitectura Modelo/Vista/Controlador.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	4.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	4.1%
4.1.x	Jelly Bean	16	13.0%
4.2.x		17	15.9%
4.3		18	4.7%
4.4	KitKat	19	39.3%
5.0	Lollipop	21	15.5%
5.1		22	2.6%



**Figura 4. Estadísticas de la fragmentación de las versiones de Android.**

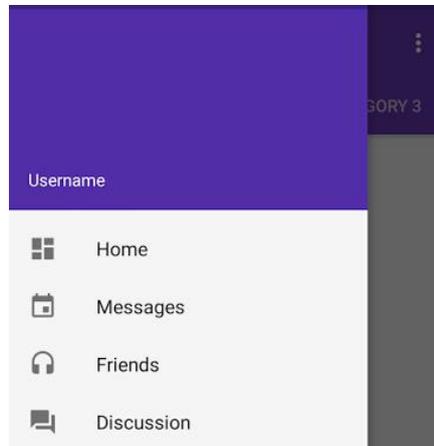
Con esta nueva librería podemos cubrir el 100% de los terminales android independizándose de la implementación para cada una de ellas. Google brinda un proyecto de ejemplo que implementado con Android Design Library que se puede ver y descargar de el siguiente repositorio git <https://github.com/hmunoz/cheesesquare>.

Al analizar el proyecto y la configuración del mismo, que se detalla en el archivo build.gradle, se puede ver la propiedad `minSdkVersion = 9`, que establece la compatibilidad superior a la versión 2.2 de Android, siendo el valor 9 la referencia al API level. También se puede observar como se agrega la dependencia de la nueva librería de la siguiente forma: `compile 'com.android.support.design:22.2.1'`, siendo 22.2.1 la versión de la SDK que estemos utilizando (Android Design Lib esta disponible de las versión 22 en adelante).

Los elementos a destacar en este proyecto que facilitan la implementación respecto a AppCompatActivity son:

### 3.2.1 Navigation View (Vista navegación)

Este componente permite implementar la navegación por nuestra aplicación de forma muy intuitiva. Todas las aplicaciones que desarrolla google tienen este tipo de navegación. Como se aprecia en la figura 5, el componente tiene la particularidad de adaptarse a distintos tamaño de pantalla. El tag para implementar este componente es el siguiente `<android.support.design.widget.NavigationView/>`.



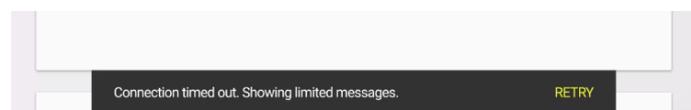
**Figura 5. Componente de navegación.**

### 3.2.2 FAB (Botón de Acción Flotante)

Es el clásico botón de acción flotante con esta nueva librería es mucho más simple implementar, solo basta utilizar el tag: `<android.support.design.widget.FloatingActionButton/>`.

### 3.2.3 Snack Bar (Barra de Notificaciones)

Componente para informar cualquier acción que se haya realizado, con la posibilidad de agregar un botón para realizar una acción complementaria. Un ejemplo podría ser: eliminar un elemento de una lista y en la Snack Bar mostrar que se eliminó y el botón de deshacer. En la figura 6, se puede ver otro ejemplo donde informa problemas de conexión y ofrece la posibilidad de reintentar. Fragmento de código para invocar snack bar es el siguiente: `Snackbar.make(this, "Connection time Out", LENGTH_LONG).setAction("Reintentar", myOnClickListener).show();`



**Figura 6. Ejemplo de snack bar**

### 3.2.4 Tabs

Ahora el componente tab tiene un aspecto renovado con animaciones de transición al cambiar entre los distintos ítems del tab padre. En la figura 7 se puede ver un ejemplo de la implementación de un tab con tres ítems, también como se aplican los estilos al componente siguiendo el concepto de Material Design.



Figura 7. Tab con tres ítems

### 3.2.5 Collapsing Toolbars (Barra de herramientas colapsable)

Este componente permite generar una barra de herramienta con animación de transición de forma simple. En la figura 8 se puede ver una transición de como cambia la imagen del contacto a medida que bajamos con el scroll, generando el efecto de transición del componente toolbar de la nueva librería Android Design Library.

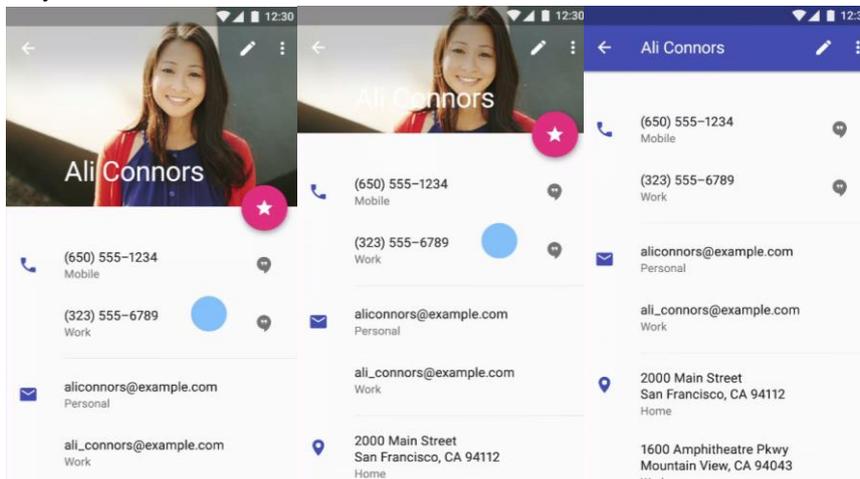


Figura 8. Efecto de Transición

El código de esta implementación se reduce a unas pocas líneas para todas las versiones de Android.

```
<android.support.design.widget.AppBarLayout >
<android.support.design.widget.CollapsingToolbarLayout
app:layout_scrollFlags="scroll|exitUntilCollapsed">
<android.support.v7.widget.Toolbar
app:layout_collapseMode="pin"/>
</android.support.design.widget.CollapsingToolbarLayout>
</android.support.design.widget.AppBarLayout>
```

## 4. Prueba de concepto

Con el propósito de comprobar que el uso de Material Design mejora el desarrollo de software en un marco ágil de trabajo, se abordó el desarrollo del proyecto “Días y Horarios de Cursada” para las carreras de la Sede Atlántica de la Universidad Nacional de Río Negro.

Personal especializado del Laboratorio de Informática Aplicada de la UNRN, estimó el tiempo de desarrollo para esta solución, utilizando para la capa de vista, la biblioteca de componentes JQueryUi del framwork JQuery, en 176 horas.

El desarrollo del software se llevó a cabo, utilizando **Angular Material**, implementando lineamientos y conceptos de Material Design, de manera de poder realizar la presente prueba de concepto. El tiempo total de desarrollo fue de 73 horas, aproximadamente en el 41,5% de tiempo estimado con JQueryUI.

En la Figura 9, podemos apreciar la pantalla de listado de materias de la carrera Licenciatura en Sistemas de la URN. Para dicha pantalla, se enlazaron, declarativamente, componentes de Material Design, los cuales permitieron obtener la barra navegacional y el listado deslizable de materias.



**Figura 9. Pantalla de Listado de Materias.**

En la figura 10, se puede observar el solapamiento de pantallas. La que se encuentra al frente, despliega el menú de opciones, producto de la implementación de componentes de material Design que representan la propiedad de “Movement of material” [13].

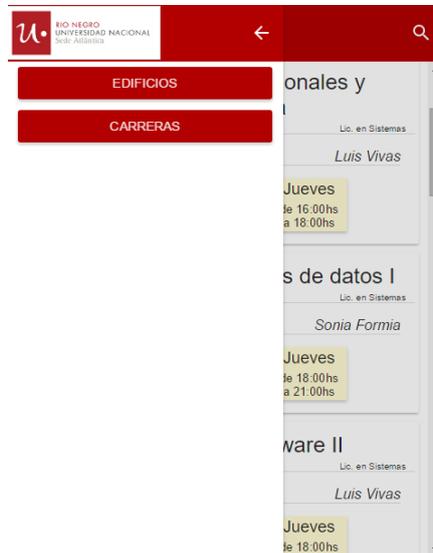


Figura 10. Despliegue del menú de opciones

## 5. Conclusión y trabajos futuros

En un marco de trabajo ágil es fundamental acotar los tiempos para poder hacer entregas tempranas del producto, una característica que requiere el mercado actual, la posibilidad de poner un producto a disposición lo antes posible para adelantarse a la competencia.

En este sentido, diseñadores, arquitectos de software y desarrolladores hacen uso de frameworks que facilitan y agilizan el proceso de construcción de software. Material Design es una herramienta que hace un aporte transversal al desarrollo de aplicaciones agregando un factor innovador. Facilita la comunicación entre todos los actores que intervienen en el desarrollo gracias a lineamientos sólidos y ampliamente aceptados.

Los fundamentos de Material Design permiten hacer posible todo lo que el diseñador desee. Los desarrolladores poseen un amplio grupo de frameworks que permiten implementar Material Design en los distintos contextos, ya sean móviles o computadoras personales.

En el desarrollo de este trabajo se expone que Material Design tiene una parte pensada para diseñadores, desarrollada por diseñadores y tecnólogos, y también, otra parte para los desarrolladores, para que el concepto y filosofía de Material Design se pueda llevar a cabo de forma sencilla y repetitiva.

También podemos decir que, si bien Polymer es el framework que permite lograr sitios y aplicaciones nunca vistos hasta hoy, también existen otros frameworks que se pueden adaptar a nuestro grupo de trabajo y lograr los objetivos de forma ágil.

En cuanto a aplicaciones nativas con Android, el aporte de Android Design Library, ayuda, sustancialmente, a implementar Material Design con menor esfuerzo y mayor compatibilidad.

Por otro lado, la prueba de concepto, arrojó un ahorro aproximado al 60% en el tiempo de desarrollo.

Los resultados obtenidos nos permiten, pensar en desarrollar líneas de trabajo futuras, que vinculen modernas herramientas y arquitecturas de software con las premisas de las metodologías ágiles.

## Referencias

1. Material Design. Google . <https://www.google.com/design/spec/material-design/> (accedido 07/15).
2. Evento I/O 2015. Google . <https://events.google.com/io2015/> (accedido 07/15).
3. Polymer. Google. <https://www.polymer-project.org/> (accedido 07/15).
4. Android Design Library. Google. <https://developer.android.com/tools/support-library/features.html> (accedido 07/15).
5. Polymer 1.0. Google Groups. <https://www.polymer-project.org/1.0/> (accedido 07/15).
6. Web-animations. W3C. <http://www.w3.org/TR/web-animations/> (accedido 07/15).
7. Angular Material. Google. <https://material.angularjs.org/latest/#/> (accedido 06/15).
8. AngularJs. Google. <https://angularjs.org/> (accedido 06/15).
9. Bootstrap. Bootstrap. <http://getbootstrap.com/> (accedido 07/15).
10. Bootstrap Material. Fezvrasta. <https://fezvrasta.github.io/bootstrap-material-design/> (accedido 07/15).
11. Polymer Starter Kit. Google. <https://developers.google.com/web/tools/polymer-starter-kit/> (accedido 06/15).
12. Fragmentación de Versiones. Google. [https://developer.android.com/about/dashboards/index.html?utm\\_source=suzunone](https://developer.android.com/about/dashboards/index.html?utm_source=suzunone) (accedido 07/15)
13. Propiedades de Material Design, Movement of Material <https://www.google.com/design/spec/what-is-material/material-properties.html#material-properties-movement-of-material>