

Chuleta de Elementos Personalizados

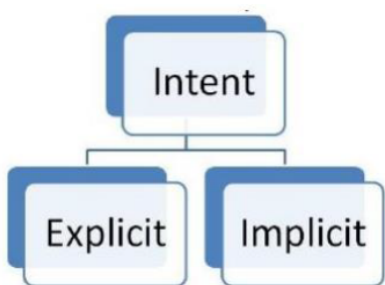
Enlaces

Será buena práctica añadir enlaces externos a las explicaciones que estamos dando en el tema. Los enlaces externos a la información se incluirán directamente en el texto, como se ve en el siguiente ejemplo:

En este tema estudiaremos los [componentes de una aplicación Android](#) y sus respectivas funciones, además de la comunicación entre activitis.

Imágenes

La mejor manera de añadir imágenes es mediante el elemento **mde_Imagen_centrada**, ya que nos permite especificar tamaño.



Explicación de código y alguna otra anotación

Llamada **chincheta** en **mde_**, la usaremos para explicar código y para comentar alguna anotación, por ejemplo:

✂ **Línea 17** con el elemento CollapsingToolbarLayout se nos permitirá realizar este efecto . En la **Línea 22** se especifica la propiedad...


Resaltar elementos del tema

Llamada **importante** en **mde_**, la usaremos para resaltar algún elemento del tema o dar algún aviso, como por ejemplo:

👉 Como en otros lenguajes de programación, se debe controlar las operaciones con variables de distintos tipos, para evitar resultados inesperados o incluso excepciones. Kotlin tiene varios métodos **toX()** para cambiar los valores al tipo que necesites...


Tareas

Para indicar alguna tarea dentro del tema usaremos el elemento personalizado **tarea**, por ejemplo.

 **Analiza el archivo de manifiesto anterior, buscando en Internet la información que no encuentres en el tema. Después haz una descripción de como crees que será la aplicación que contiene ese manifiesto.**

Ejemplos

con el `_mde` llamado ejemplos, construiremos algún ejemplo que queramos incluir en el tema:

 Vamos a definir nuestra aplicación para poder configurarla en tres idiomas: español, inglés y francés, para una aplicación multi-idioma sería necesario incluir recursos de tipo cadena para cada idioma que tenga la aplicación, para ello crearíamos tres versiones del fichero `strings.xml` y lo guardaríamos en los recursos adicionales de valores según el idioma, para saber las extensiones de los [idiomas](#), aunque Android Studio nos proporciona la opción de crear recursos fácilmente. Por ejemplo, si dentro de la carpeta `res`, seleccionas **new -> Android Resources File** y seleccionamos `value`, podremos crear los recursos adicionales por localidad:

Bloque Kotlin

Para añadir código de Kotlin usaremos `mde_bloque_kotlin`, que nos permite señalar líneas.

1

```
if (intento.resolveActivity(packageManager) != null) startActivity(intento);
```

Bloque Xml

Para añadir código en xml usaremos `mde_bloque_xml`, que nos permite señalar líneas.

0

```
<action android:name="android.intent.action.VIEW"/>
<category android:name="android.intent.category.DEFAULT"/>
<data android:mimeType="vnd.android.cursor.dir/person"/>
</intent-filter>
```

Saltos de página

En ocasiones conviene añadir un salto de página para que el pdf quede más correcto. Se llama `mde_saltoPagina`


Para resaltar alguna idea


Usaremos el **mde_cita_tips**

 **Tips:**

Buenas y malas prácticas

Si queremos indicar alguna cosa como buena o mala practica, usaremos los estilos **mde_icono_buena_practica** y **mde_icon_mala_practica**

✖  Esto es una mala práctica

✔  Esto es una buena práctica