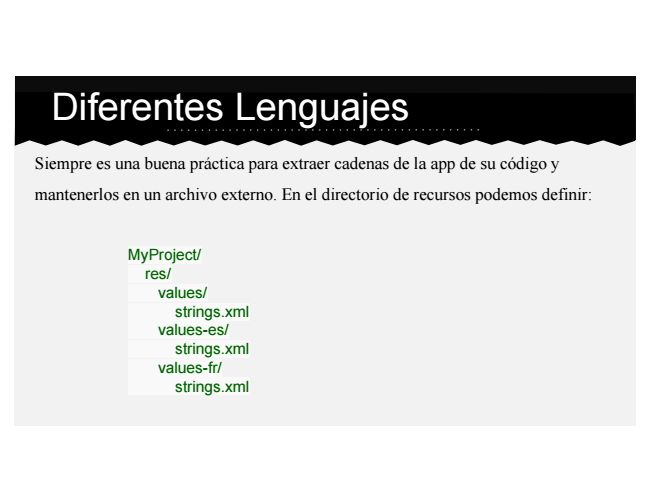


*Flexibilidad de App Android en idioma, def. pantallas, ver. plataforma.*

**

Diferentes Lenguajes

Siempre es una buena práctica para extraer cadenas de la app de su código y

mantenerlos en un archivo externo. En el directorio de recursos podemos definir:

MyProject/

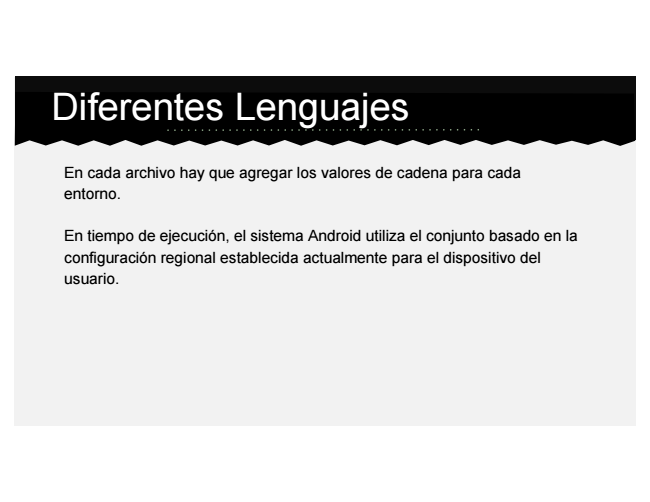
res/

values/

strings.xml values-es/

strings.xml values-fr/

strings.xml



Diferentes Lenguajes

En cada archivo hay que agregar los valores de cadena para cada entorno.

En tiempo de ejecución, el sistema Android utiliza el conjunto basado en la configuración regional establecida actualmente para el dispositivo del usuario.



Diferentes Lenguajes

Por ejemplo, las siguientes son algunas diferentes archivos de recursos de String en diferentes idiomas.

Inglés (por default), /values-es/strings.xml: <?xml version="1.0" encoding="utf-8"?> <resources>

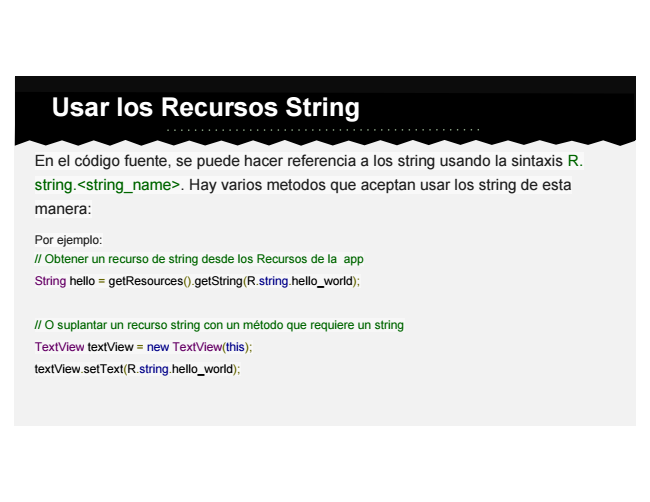
<string name="title">My Application</string> <string name="hello\_world">Hello World!</string> </resources>

Español, /values-es/strings.xml: <?xml version="1.0" encoding="utf-8"?> <resources>

<string name="title">Mi Aplicación</string> <string name="hello\_world">Hola Mundo!</string> </resources>

Frances, /values-fr/strings.xml: <?xml version="1.0" encoding="utf-8"?> <resources>

<string name="title">Mon Application</string> <string name="hello\_world">Bonjour le monde !</string> </resources>



**Usar los Recursos String**

En el código fuente, se puede hacer referencia a los string usando la sintaxis R. string.<string\_name>. Hay varios metodos que aceptan usar los string de esta manera:

Por ejemplo: // Obtener un recurso de string desde los Recursos de la app

String hello = getResources().getString(R.string.hello\_world);

// O suplantar un recurso string con un método que requiere un string

TextView textView = new TextView(this);

textView.setText(R.string.hello\_world);



**Usar los Recursos String**

En otros archivos XML, se puede hacer referencia a un string con la sintaxis @string/<string\_name> cuando el atributo XML acepta ese valor.

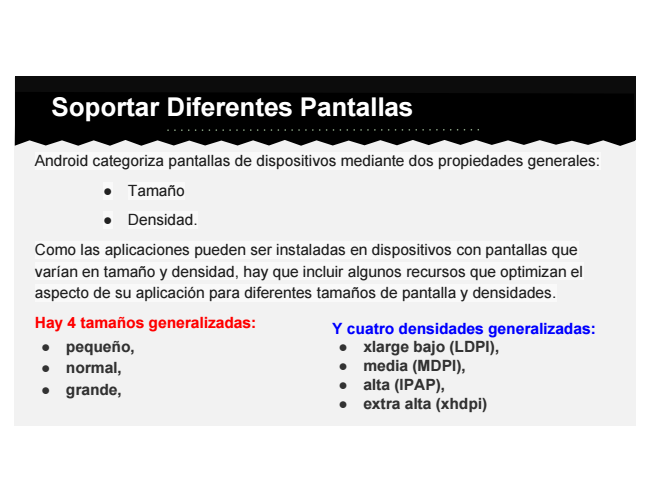
Por ejemplo:

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/hello\_world" />



**Soportar Diferentes Pantallas**

Android categoriza pantallas de dispositivos mediante dos propiedades generales:

● Tamaño

● Densidad.

Como las aplicaciones pueden ser instaladas en dispositivos con pantallas que varían en tamaño y densidad, hay que incluir algunos recursos que optimizan el aspecto de su aplicación para diferentes tamaños de pantalla y densidades.

**Hay 4 tamaños generalizadas:**

**● pequeño,**

**● normal,**

**● grande,**

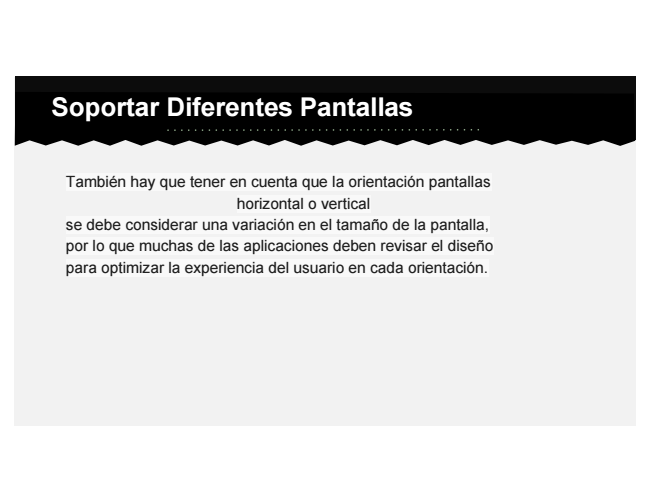
**Y cuatro densidades generalizadas:**

**● xlarge bajo (LDPI),**

**● media (MDPI),**

**● alta (IPAP),**

**● extra alta (xhdpi)**

****

**Soportar Diferentes Pantallas**

También hay que tener en cuenta que la orientación pantallas

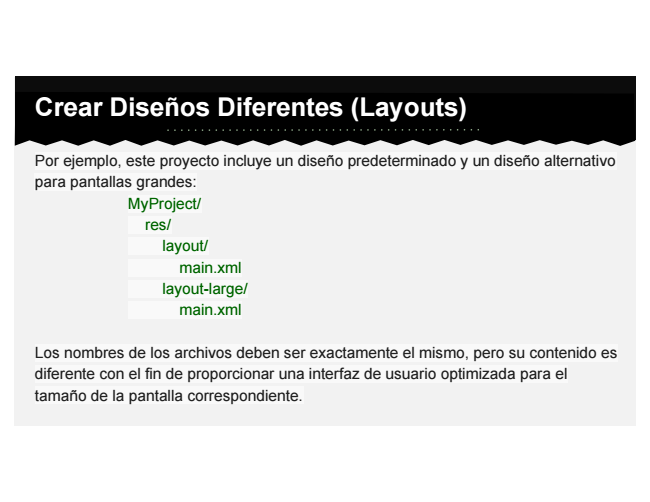
horizontal o vertical se debe considerar una variación en el tamaño de la pantalla, por lo que muchas de las aplicaciones deben revisar el diseño para optimizar la experiencia del usuario en cada orientación.



**Crear Diseños Diferentes (Layouts)**

Se debe crear un archivo de formato XML único para cada tamaño de pantalla que desea soportar. Cada diseño debe ser guardado en el directorio de recursos apropiados, nombrada con un sufijo -<screen\_size>. Por ejemplo, un diseño único para pantallas grandes se debe guardar bajo res/layout-large/.

Importante: Android escala automáticamente la disposición con el fin de encajar correctamente en la pantalla. Así que los diseños para diferentes tamaños de pantalla no tienen que preocuparse por el tamaño absoluto de los elementos de la interfaz de usuario, sino que se centran en la estructura de disposición (como el tamaño o la posición en relacion a los puntos de vista importantes con hermanos y visitas).



**Crear Diseños Diferentes (Layouts)**

Por ejemplo, este proyecto incluye un diseño predeterminado y un diseño alternativo para pantallas grandes:

MyProject/

res/

layout/

main.xml layout-large/

main.xml

Los nombres de los archivos deben ser exactamente el mismo, pero su contenido es diferente con el fin de proporcionar una interfaz de usuario optimizada para el tamaño de la pantalla correspondiente.