



El futuro digital
es de todos

MinTIC

Unidad 3

Aplicaciones móviles



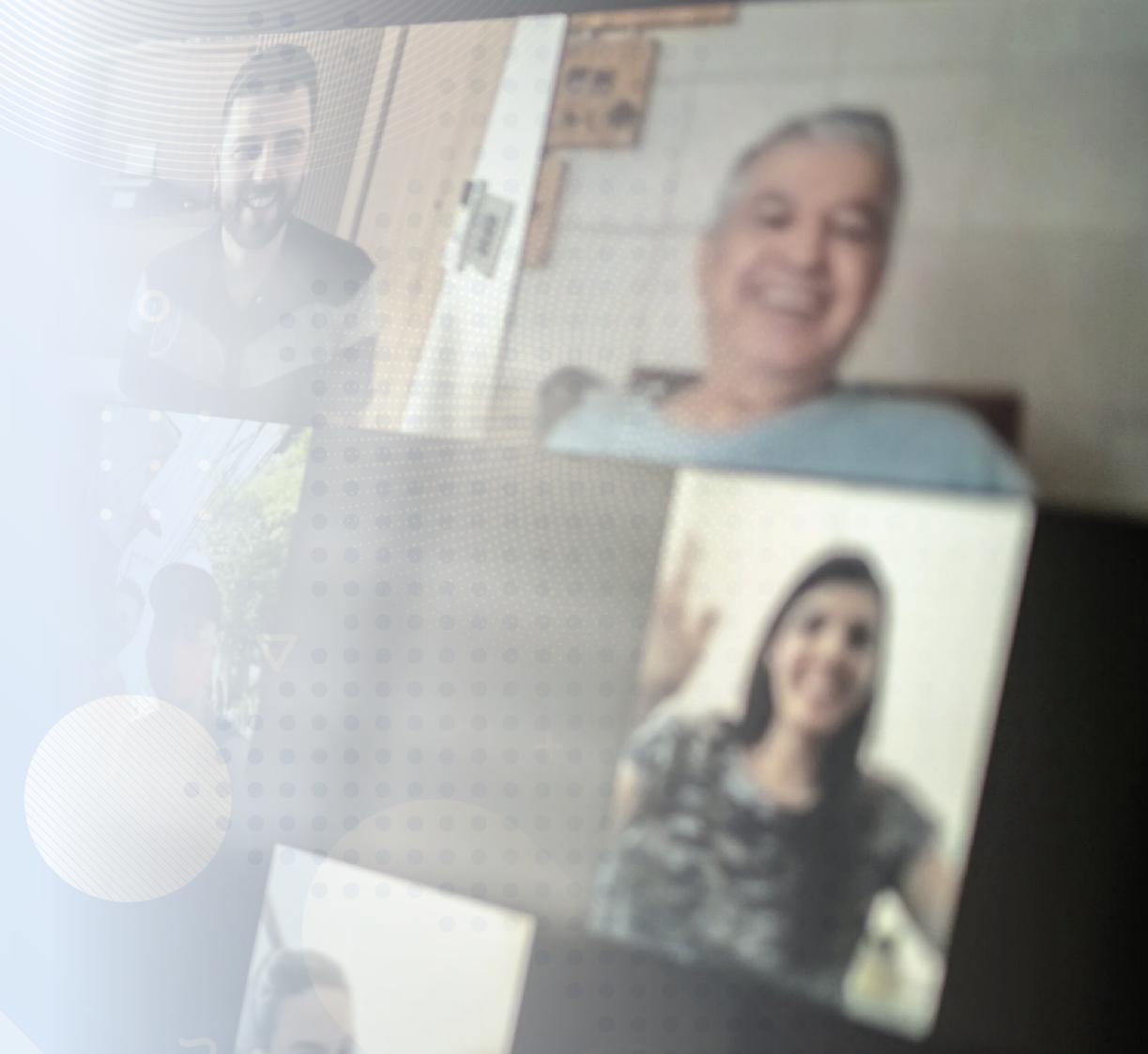


El futuro digital
es de todos

MinTIC

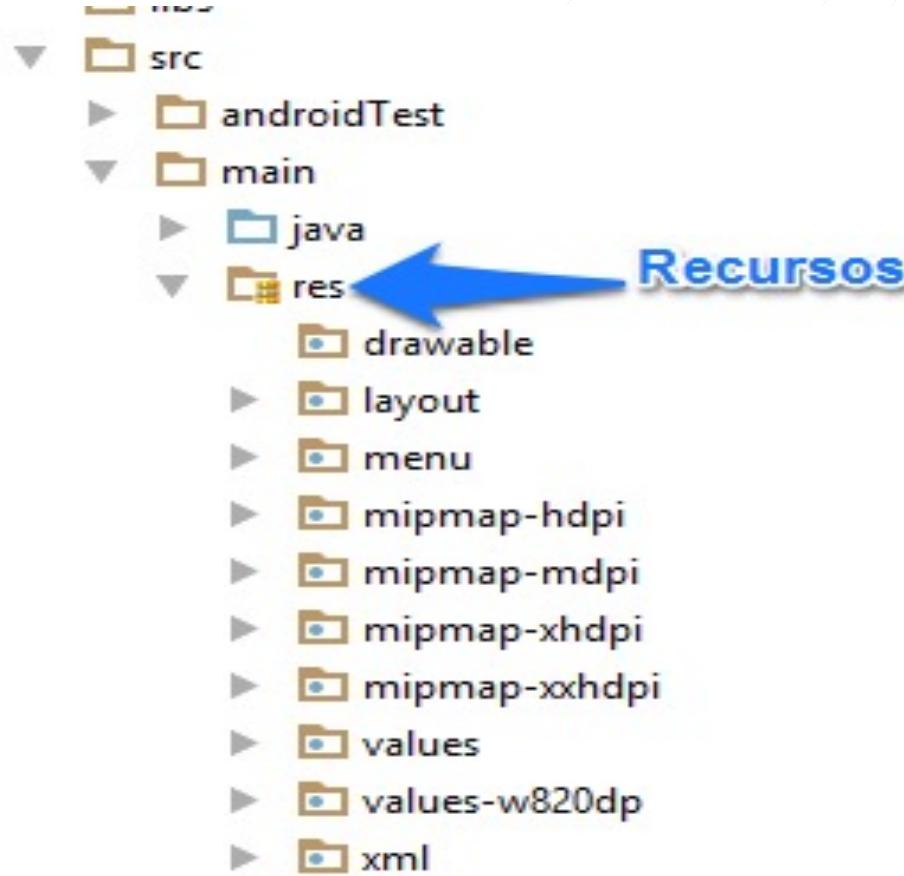
Tema 5

Uso de recursos en Android





Uso de recursos en Android



Los recursos son archivos o datos externos que soportan el comportamiento de nuestra aplicación Android. Es decir, imágenes, strings, colores, estilos, etc.

Formalmente en un proyecto, estos elementos se encuentran en la carpeta src/main/res. Allí se encuentran subdirectorios que agrupan los diferentes tipos de recursos.



Uso de recursos en Android



La idea del uso de recursos es dividir el código de la app para mantener independencia. Todo con el fin de agregar variaciones de los archivos para adaptar la aplicación a diferentes tipos de pantallas, idiomas, versiones, dimensiones, configuraciones de orientación, etc

Por ejemplo



No es lo mismo el espacio de un teléfono móvil a una tableta. Sin embargo, se pueden crear variaciones de los recursos para que la aplicación se adapte a la densidad de pantalla de cada uno.



Tipos de recursos

Los grupos de recursos se dividen en subdirectorios. Cada uno de ellos contiene archivos que cumplen una función específica dentro de la aplicación. Se debe respetar esta estructura de documentos para no tener problemas en la ejecución.

La siguiente es una tabla de la mayoría de los tipos de recursos que se pueden crear:



Tipos de recursos

Nombre del subdirectorio	Contenido
animator/	Alberga archivos XML con animaciones de propiedades para objetos.
anim/	Contiene archivos XML que representan animaciones especiales para views.
color/	Aquí encontrarás archivos XML con definiciones de listas de estados de color (Color State List). Estas listas determinan el color de un componente dependiendo del estado en que se encuentre.
drawable/	Recursos gráficos que puedan ser proyectados en pantalla. Generalmente encontrarás archivos de imagen como .png , .jpg o .gif , sin embargo es posible usar otros como: archivos nine-patch, listas de estado, drawables con múltiples niveles, drawables con figuras 2D definidas en XML, y muchas más.



Tipos de recursos

Nombre del subdirectorio	Contenido
mipmap/	Contiene el icono de la aplicación para evitar distorsión entre varias densidades de pantalla.
layout/	Archivos XML que contienen definiciones de la interfaz de usuario.
menu/	Archivos XML que establecen las características para los menús usados en la interfaz. Normalmente contienen definiciones sobre los ítems albergados en un menú y las agrupaciones entre ellos.
raw/	Almacena aquí todos aquellos archivos que deseas leer directamente como un flujo de caracteres estándar (InputStream).



Tipos de recursos

Nombre del subdirectorio	Contenido
values/	<p>Archivos XML que contienen datos simples como enteros, strings, booleanos, colores. Puedes agrupar estos elementos en las siguientes convenciones de archivos:</p> <ul style="list-style-type: none">• strings.xml para cadenas• colors.xml para definiciones de colores en hexadecimal.• dimens.xml para dimensiones de márgenes, padding, tamaños, etc.• styles.xml para los estilos de interfaz que tendrá la aplicación.• arrays.xml para arreglos de elementos.
xml/	Contiene archivos XML con definiciones especiales requeridas en acciones con el framework de Android. O para usos que requieras en tus requerimientos, como parsear una estructura XML directamente en la aplicación.



Recursos String en Android

Los recursos de tipo strings permiten almacenar valores que representan texto en la aplicación.



Los elementos más comunes son strings simples con un solo contenido y arrays de strings que asocian varios elementos a un solo nombre.

Son de gran utilidad, ya que se pueden crear variaciones de idioma para cada recurso string.

El archivo **values/strings.xml** debe ser designado por convención para contener este tipo de recursos.



Strings y Arrays

Los strings simples se definen con la etiqueta «string». Para darle un identificador se usa el atributo name y se asigna un nombre distintivo

```
<string name="nombre_string">texto_string</string>
```

Los arrays se declaran con «string-array» y asignarles un nombre con el atributo name. Cada elemento irá en su interior declarado con un elemento «item» y su contenido será el texto que representa.



Strings y Arrays

```
<resources>
    <!-- strings de contenido simple -->
    <string name="app_name">Mis Pruebas</string>
    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>

    <!-- Array de strings -->
    <string-array name="campeones_adc">
        <item>Caitlyn</item>
        <item>Draven</item>
        <item>Jinx</item>
        <item>Varus</item>
        <item>Tristana</item>
    </string-array>
</resources>
```

Este código muestra en la parte superior tres strings simples y en su parte inferior un array de strings con 5 ítems.



Strings

Una ventaja de usar Android Studio es que permite extraer textos hacia el archivo strings.xml a través de un asistente.

Si se estuviese definiendo un layout donde se encuentra el siguiente TextView:

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:text="Descripción de tareas"/>
```

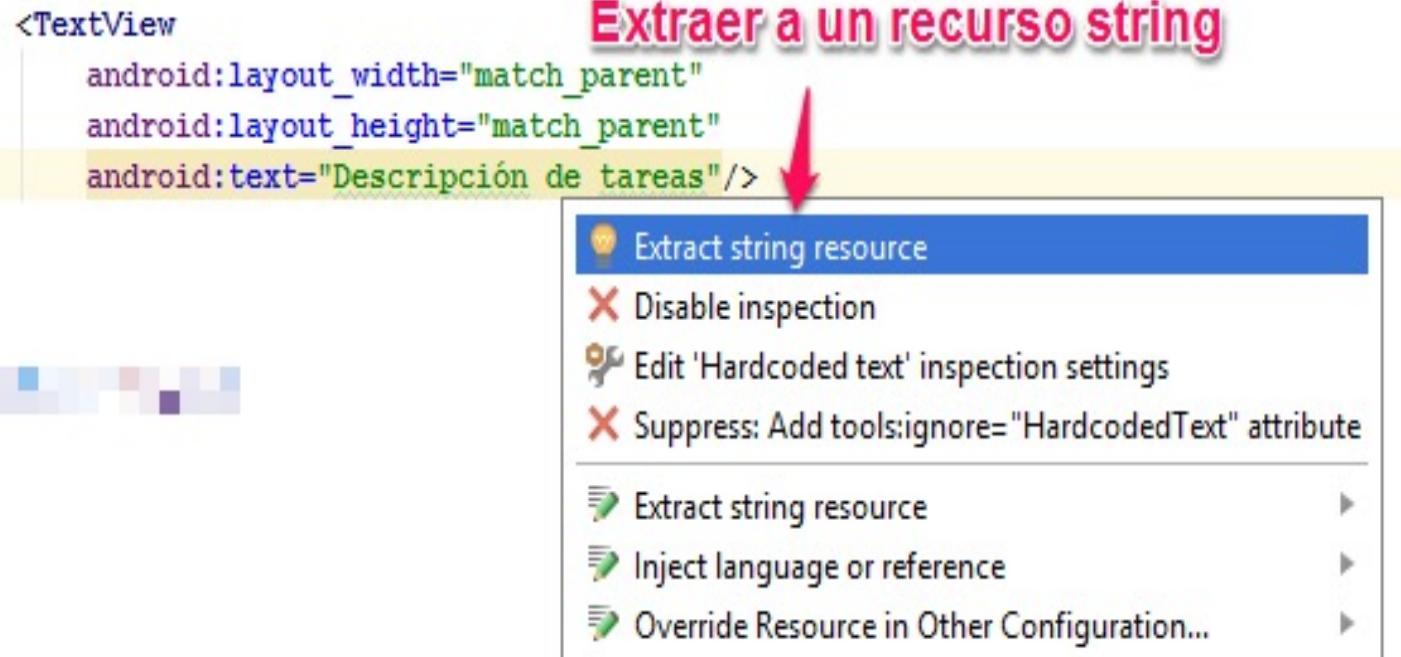
Se verá que el valor del atributo android:text aparecerá marcado.

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:text="Descripción de tareas" />
```





Strings archivo xml

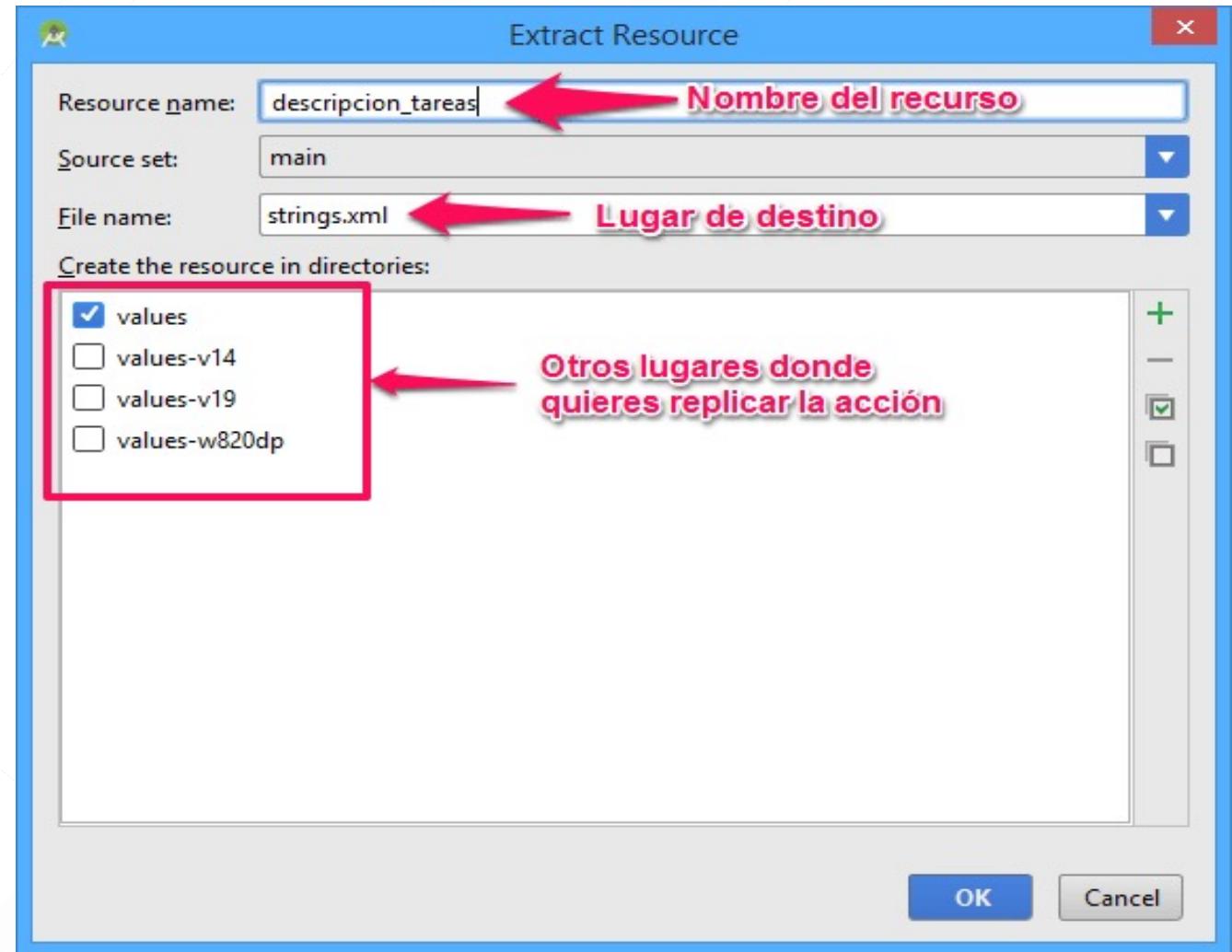


Este warning está diciendo que sería buena idea extraer ese texto plano a un recurso string. Si se presiona ALT+ENTER en esa posición, aparecerá una lista de opciones. La que se debe usar es Extract string resource.



Strings archivo xml

Si seleccionas esta opción se desplegará un asistente que nos permitirá trasladar el texto plano al archivo strings.xml con su respectivo identificador.





Strings archivo xml

Una vez se asigne el nombre, se confirma y ya estará el nuevo recurso creado. Incluso el acceso por el text view será reemplazado.

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:text="@string/descripcion_tareas"/>
```