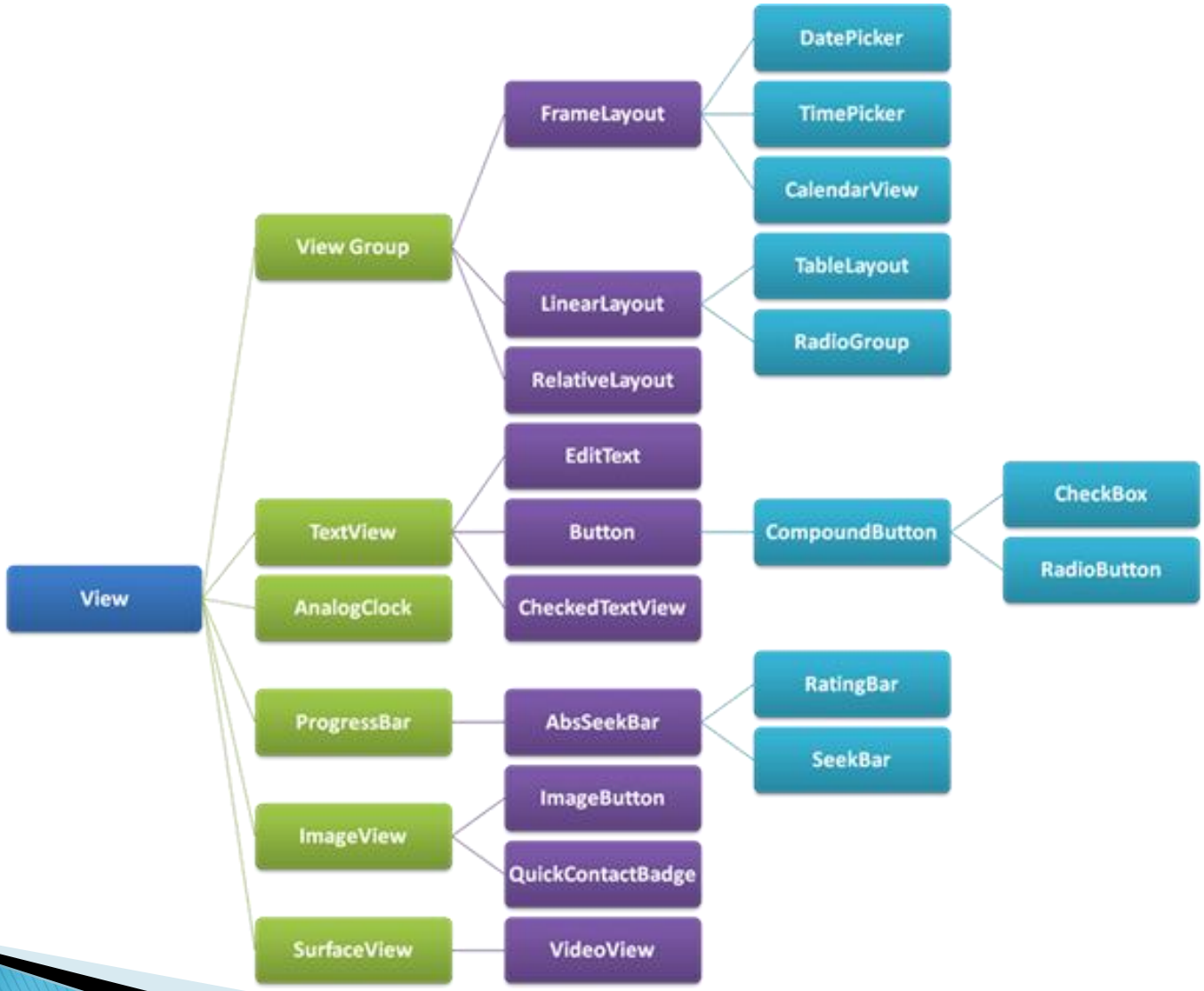
Interfaz de usuario

Mª Alejandra Tomás Fernández

Interfaz de usuario (I)

- La interfaz gráfica de usuario para una aplicación Android está construida mediante una jerarquía de **View** y objetos **ViewGroup**.
- Los objetos **View** son generalmente los elementos que se ven en la pantalla y con los que *interactúa el usuario*:
  - ***Botones***
  - ***Cuadros de texto***
  - ***Checkbox***, …
  - Estos controles con los que interactúa el usuario, en Android, se les llama **WIDGETS**.
  - La **clase View** es la clase base de todos los **WIDGETS**: *Button, EditText*,…

Interfaz de usuario (II)



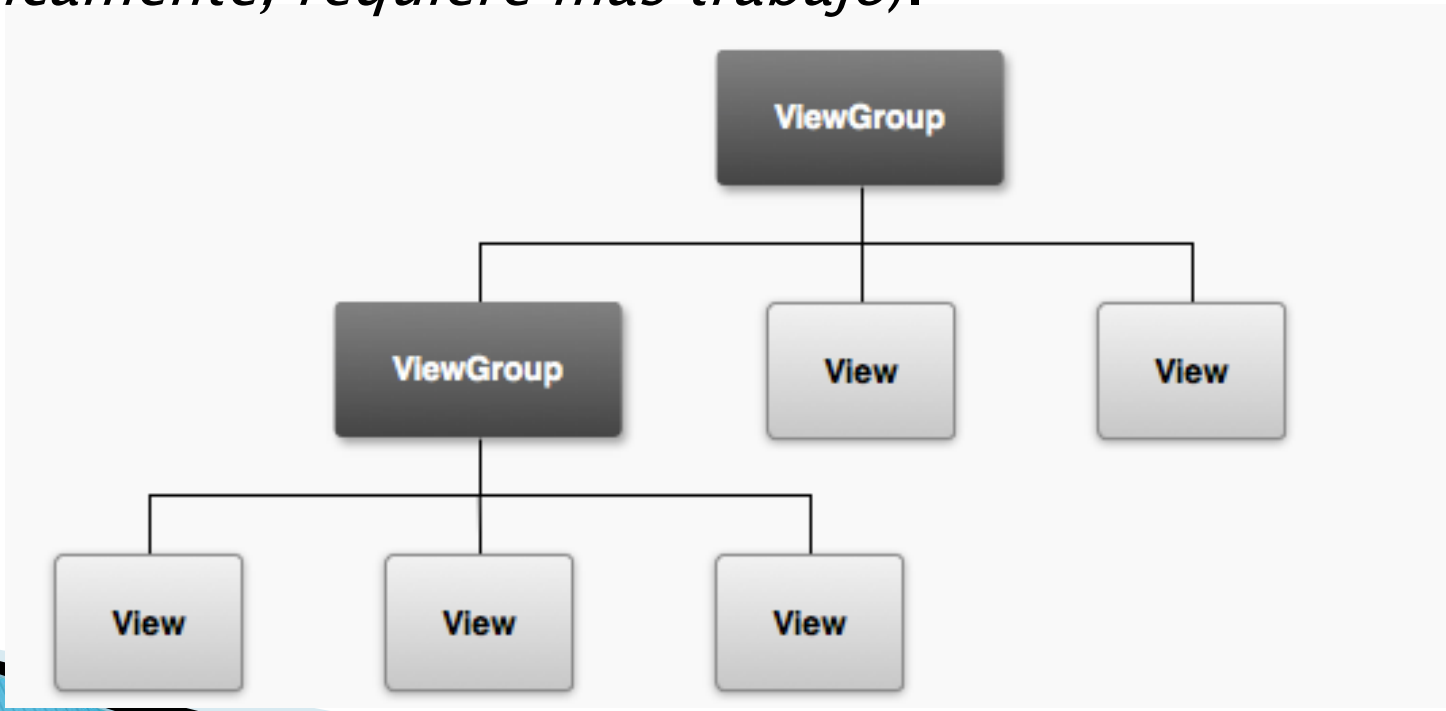
Mª Alejandra Tomás Fernández

Interfaz de usuario (III)

- Los objetos **ViewGroup** son *contenedores invisibles* que controlan la *distribución*, *posición* y *dimensiones de los controles* que se insertan en su interior:
  - Pueden *disponerse en una rejilla*,
  - En una **tabla**,
  - En una **lista vertical**,
  - **Disposición lineal** (*LinearLayout)*, ...
- Un **ViewGroup** es un **View especial** que **puede contener otros View** (*llamados hijos*): que pueden ser **View** o **ViewGroup**.
- **El ViewGroup es la clase base para los "layouts" y "otras vistas contenedoras".**

Interfaz de usuario (III)

- Así la **jerarquía de vistas que definen el diseño de la interfaz de usuario** será algo como:
- **Se pueden hacer subclases de View** y de **ViewGroup para crear nuestras propias vistas y widgets** (*lógicamente, requiere más trabajo*).



Interfaz de usuario (V): Formas de crear la Interfaz de Usuario (I)

- **Hay 3 formas:**

1. Mediante lenguaje XML:

- La **forma más habitual** de **definir un Layout** es mediante un archivo de Layout en **XML**, que se guarda en */src/main/res/layout:*
  - Permite **separar el diseño de la interfaz del código** que lo maneja.
  - Además, **se pueden crear configuraciones alternativas para diferentes tamaños de pantalla, lenguajes, orientaciones…**
  - De nuevo, **se puede establecer qué Layout usar con setContentView()** y pasando el ID del layout, situado en R.
- El nombre de un **elemento XML** (*etiqueta*) para una determinada vista corresponde al **nombre de la clase** que lo representa en **Java**.
- Así el **elemento <TextView>** creará **un widget "Textview"**, un elemento **<LinearLayout>** creará un **ViewGroup "LinearLayout"**, …
- Cuando se carga un **recurso layout**, (*carpeta /src/main/res/layout*) el sistema Android ***inicializa cada nodo del layout como un objeto en tiempo de ejecución.***
- Para ver **ejemplos de Layout XML** podemos referirnos a los ejercicios ya hechos.

Interfaz de usuario (VI):
Formas de crear la Interfaz de Usuario (II)

2. **Mediante código JAVA:**

   - Esta forma **permite crear y/o modificar la interfaz en tiempo de ejecución**.

   - A veces, es útil poder realizar cambios en tiempo de ejecución.

3. **Usando la herramienta de diseño gráfico que provee Android Studio:**

   - Es más intuitivo, pero menos exacto

Layouts mediante XML: (II)
Cargar el XML

```
public void onCreate(Bundle savedInstanceState){
  super.onCreate(savedInstanceState);
  setContentView(R.layout.main_layout);
```

Layouts mediante XML: (III)
Atributos comunes: ID

- Un atributo común a todas las Views es "**ID**".

```
android:id="@+id/un_boton"
```

- La **@ indica que el parser XML debe interpretar y expandir el resto de la cadena** (*hasta el fin de las comillas*) **e identificarlo como un recurso**.
- El símbolo **+ indica que este es un nuevo nombre de recurso que debe ser creado y añadido a los recursos** (en R).

Layouts mediante XML: (IV)
Atributos comunes: ID

- Hay otros ID de recursos pertenecientes a Android, que nos ofrece el framework. Cuando hagamos referencia a un ID de recurso de Android, no se necesita el +, pero se necesita el espacio de nombres del paquete android:

```
android:id="@android:id/empty"
```

Layouts mediante XML: (V)
Atributos comunes: ID

- Es importante distinguir esto:

  - Unos **ID de recurso son locales**, que son de nuestra aplicación.
    - Estos se **compilan en R**.
  - **Otros son de android**, a los que habrá que anteponer el espacio de nombres "android".
    - Estos se **compilan en android.R.**