

Desarrollo de apps con Android



7. RecyclerView

¿Qué es un RecyclerView?

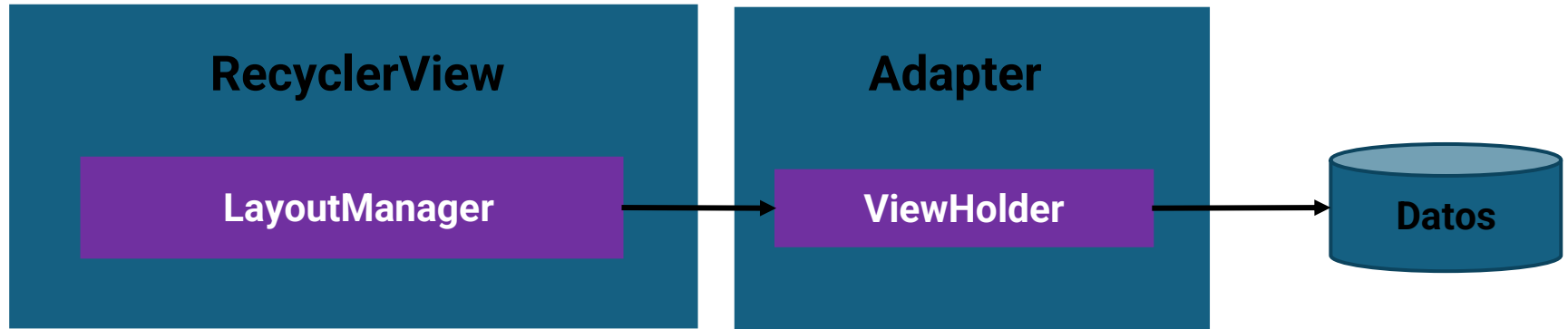
- RecyclerView es un contenedor deslizable (scroll) que permite mostrar listas de datos.
- Eficiente:
 - Utiliza y reutiliza un número limitado de elementos View.
 - Actualiza los cambios en los datos de una manera muy rápida.



RecyclerView

- **Datos**
- Lista deslizable **RecyclerView** para listas de elementos—[RecyclerView](#)
- **Layout** para un elemento de datos—fichero XML.
- **Layout manager** gestiona la organización de los componentes IU en una View—[Recyclerview.LayoutManager](#)
- **Adapter** conecta los datos al RecyclerView—[RecyclerView.Adapter](#)
- **ViewHolder** tiene la información para mostrar un elemento en una View—[RecyclerView.ViewHolder](#)

Relación entre los componentes

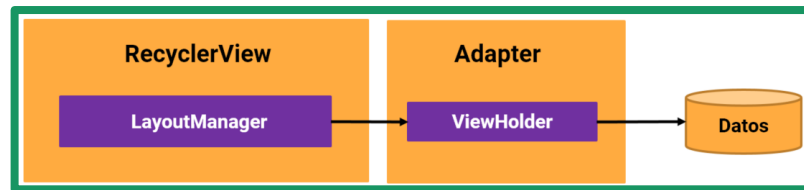


¿Qué es layout manager?

- Se utiliza para posicionar los elementos de una View dentro de un [RecyclerView](#)
- Reutiliza los elementos de la View que ya no son visibles al usuario.
- Layout managers predefinidos:
 - [LinearLayoutManager](#)
 - [GridLayoutManager](#)
 - [StaggeredGridLayoutManager](#)

¿Qué es ViewHolder?

- Utilizado por el adapter para preparar una View con los datos de un elemento de la lista.
- Layout especificado en un fichero de recursos XML.
- Puede contener elementos sobre los que se puede hacer clic.
- Se posiciona a partir del layout manager.
- [RecyclerView.ViewHolder](#)



Implementación de un RecyclerView

Pasos a seguir

1. Añadir la dependencia de RecyclerView al fichero **build.gradle.kts (módulo app)**.
2. Añadir el componente RecyclerView al layout.
3. Crear el layout XML layout para cada elemento del RecyclerView.
4. Crear clase que herede de RecyclerView.Adapter.
5. Crear clase que herede de RecyclerView.ViewHolder.
6. En el método onCreate() de la Activity, crear el RecyclerView con el adapter y el layout manager.

Añadir dependencia en build.gradle.kts (app)

```
dependencies {  
    implementation(libs.appcompat)  
    implementation(libs.material)  
    implementation(libs.activity)  
    implementation(libs.constraintlayout)  
    implementation(libs.androidx.recyclerview)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.ext.junit)  
    androidTestImplementation(libs.espresso.core)  
}
```

Crear layout con el RecyclerView

content_main.xml:

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

activity_main.xml:

```
...  
<include android:id="@+id/content" layout="@layout/content_main" />  
...
```

Crear layout para 1 elemento de la lista

word_list_element.xml:

```
<androidx.appcompat.widget.LinearLayoutCompat
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">
    >
    <TextView
        android:id="@+id/word"
        style="@style/word_title" />
</androidx.appcompat.widget.LinearLayoutCompat>
```



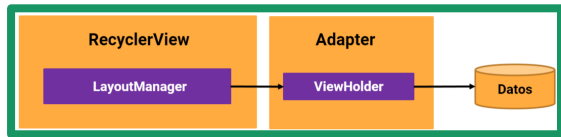
Implementar el Adapter

```
public class WordListAdapter
    extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {
    protected LinkedList<String> mWordList;
    protected View.OnClickListener onClickListener;

    public WordListAdapter(LinkedList<String> wordList) {
        this.mWordList = wordList;
    }

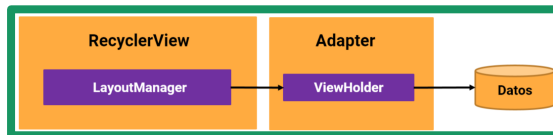
    public void setOnItemClickListener(View.OnClickListener onClickListener) {
        this.onClickListener = onClickListener;
    }

    ...
}
```



Crear ViewHolder en la clase Adapter

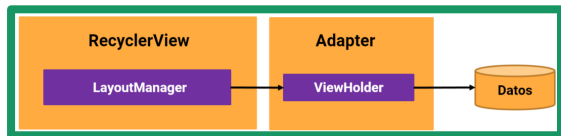
```
public static class WordViewHolder extends RecyclerView.ViewHolder {  
    public TextView wordItemView;  
  
    ...  
}
```



Constructor del ViewHolder

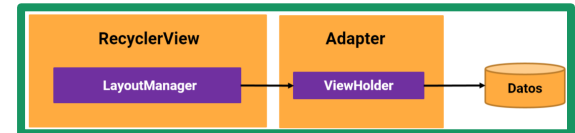
```
(A) public WordViewHolder(WordListElementBinding itemView) {  
    super(itemView.getRoot());  
    wordItemView = itemView.word;  
}
```

```
(B) public WordViewHolder(View view) {  
    super(view);  
    wordItemView = view.findViewById(R.id.word);  
}
```



Adapter requiere 3 métodos

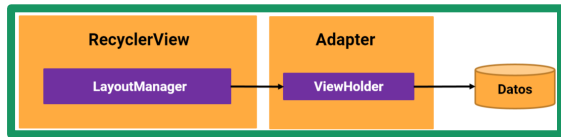
- `onCreateViewHolder()`
- `inBindViewHolder()`
- `getItemCount()`



Adapter: onCreateViewHolder() (1/2)

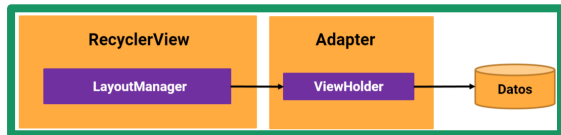
@Override

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    {  
        // Cargar la vista desde el xml (con binding)  
        WordListElementBinding v = WordListElementBinding.inflate(LayoutInflater  
            .from(parent.getContext()), parent, false);  
        // Activar el listener para el evento clic.  
        // En la clase MainActivity hay que activar este listener.  
        v.getRoot().setOnClickListener(onClickListener);  
        return new ViewHolder(v);  
    }  
}
```



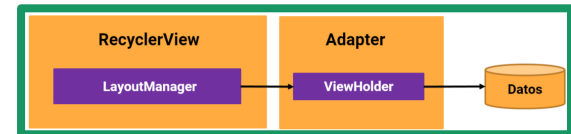
Adapter: onCreateViewHolder() (2/2)

```
@Override  
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    {  
        // Cargar la vista desde el xml (con View)  
        View view = LayoutInflater.from(parent.getContext())  
            .inflate(R.layout.word_list_element, parent, false);  
        return new ViewHolder(view);  
    }  
}
```



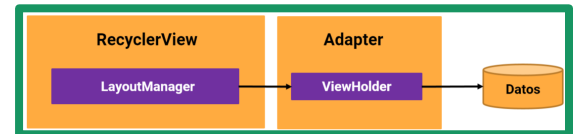
Adapter: onBindViewHolder()

```
@Override  
public void onBindViewHolder(  
    WordViewHolder holder, int position) {  
    // Recuperar los datos de esa posición.  
    String mCurrent = mWordList.get(position);  
    // Añadir el dato a la view.  
    holder.wordItemView.setText(mCurrent);  
}
```



Adapter: getItemCount()

```
@Override  
public int getItemCount() {  
    // Devolver el número de elementos a mostrar.  
    return mWordList.size();  
}
```



Crear el RecyclerView en Activity onCreate()

```
public class MainActivity extends AppCompatActivity {  
    private WordListAdapter mAdapter;  
    ...  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        mAdapter = new WordListAdapter(mWordList); //se debe tener mWordList  
        RecyclerView recyclerView = binding.content.recyclerView;  
        recyclerView.setHasFixedSize(true);  
        recyclerView.setLayoutManager(new LinearLayoutManager(this));  
        recyclerView.setAdapter(adapter);  
        // Controlar cuando se hace clic.  
        adapter.setOnItemClickListener(v -> {  
            int pos = binding.content.recyclerView.getChildAdapterPosition(v);  
            //A continuación, invocar al método que responda al evento de haber  
            // hecho clic sobre el elemento de la lista que está en la posición pos.  
        });  
    }  
}
```

