Pantallas en proyecto Android

Competencia General

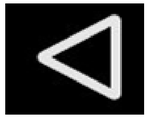
Construir una aplicación Android con la que el usuario pueda interactuar y navegar, utilizando las vistas requeridas de acuerdo a la especificación entregada.

MODULOS

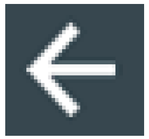
Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Navegación back (temporal)
Proporcionada por el botón Atrás del dispositivo
Controlada por la pila de respaldo del sistema Android



Navegación ancestral (arriba)
Botón arriba proporcionado en la barra de la aplicación
Controlada definiendo la actividad principal para la actividad secundaria en AndroidManifest.xml



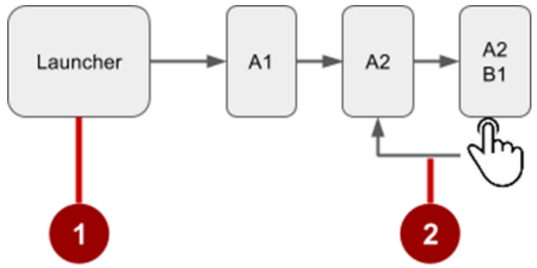
MODULOS

Las pantallas en un proyecto Android

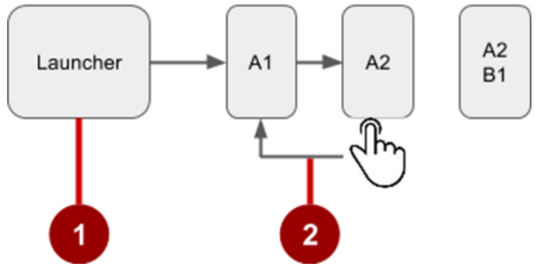
Navegación de usuario; Dos formas de navegación

Navegación por el historial de pantallas

Historys comienza desde Launcher



El usuario hace clic en el botón Atrás para navegar a las pantallas anteriores en orden inverso

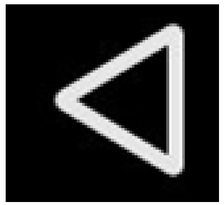


MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

El comportamiento del botón back se resumiría en algo así:



- El sistema Android administra la pila de actividades y el botón Atrás
- Si tienes dudas, no cambies
- Solo anule, si es necesario para satisfacer las expectativas del usuario

Por ejemplo: en un navegador integrado, activa el comportamiento de retroceso predeterminado del navegador cuando el usuario presiona el botón Atrás del dispositivo

Esto en código es:

```
@Override
public void onBackPressed() {
    // Add the Back key handler here.
    return;
}
```

MODULOS

Las pantallas en un proyecto Android: Activity

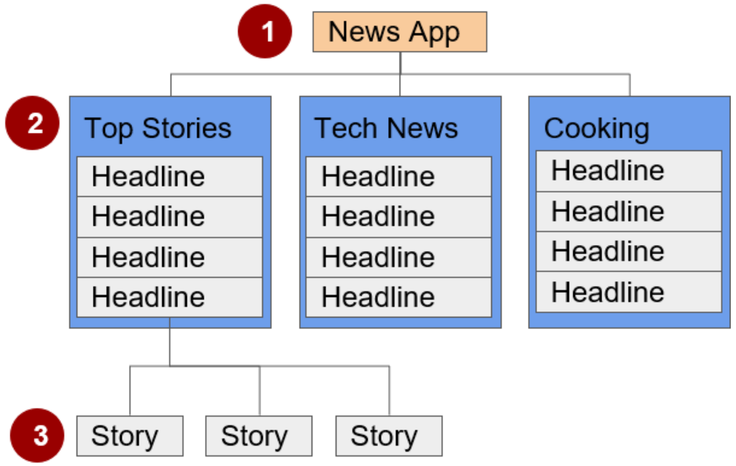
Patrones de navegación jerárquica

Pantalla principal: pantalla que permite la navegación hacia las pantallas secundarias, como la pantalla de inicio y la actividad principal.

Collection sibling: pantalla que permite la navegación a una colección de pantallas secundarias, como una lista de titulares

Section sibling: pantalla con contenido, como una historia

En un Ejemplo de una jerarquía de pantallas se tiene el esquema de la figura



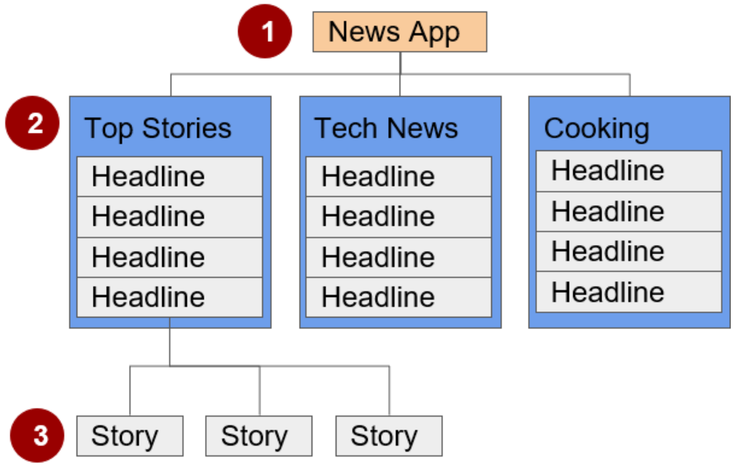
MODULOS

Las pantallas en un proyecto Android: Activity

Patrones de navegación jerárquica

1. Padre
2. Hija: collection siblings
3. Hija: section siblings

Usar Activity para la la pantalla padre
Usar Activity o Fragment para las pantallas hijas



MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Navegación descendiente

- De una pantalla principal a una de sus secundarias
- Desde una lista de titulares, hasta un resumen de la historia, hasta una historia.

Navegación ancestral

- De la pantalla de “child” o una “sibling” hermana, a su pantalla padre
- De un resumen de la historia a los titulares

Navegación lateral

- De una pantalla hermana a otra hermana
- Deslizarse entre vistas con pestañas

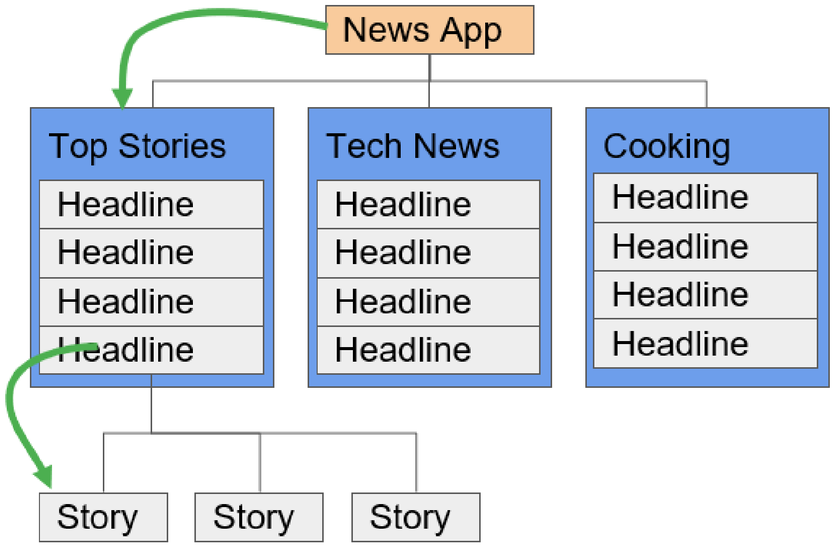
MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Navegación descendiente

- De una pantalla principal a una de sus secundarias
- De la pantalla principal a una lista de titulares a una historia



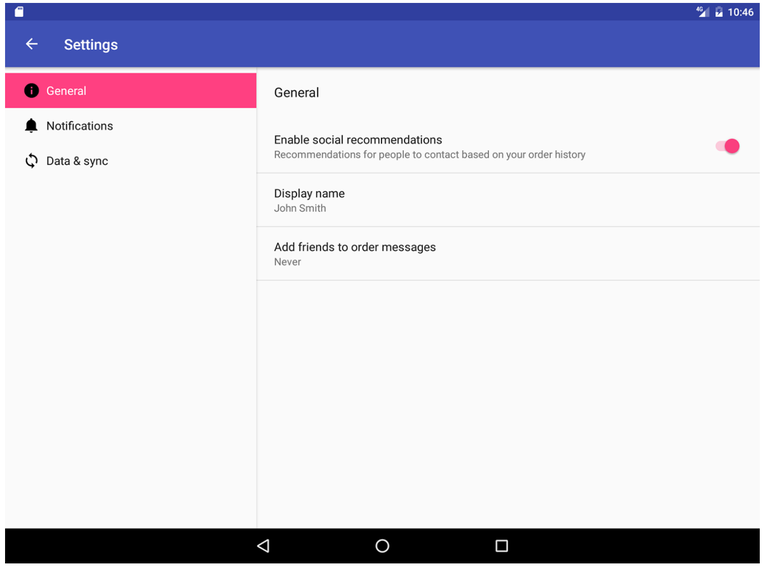
MODULOS

Las pantallas en un proyecto Android

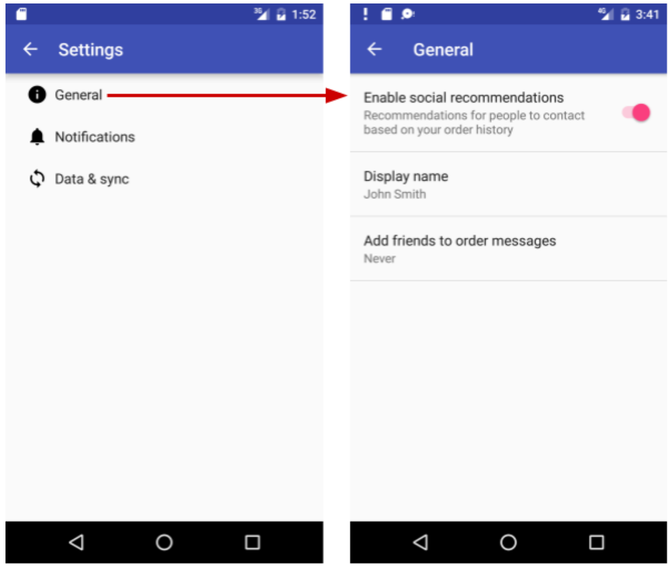
Navegación de usuario; Dos formas de navegación

Navegación descendiente

Implementación side- by-side en Tablet



En múltiples pantallas en teléfono.



MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Navegación descendiente

Controles para la navegación descendiente

- “Cajón” de navegación (Navigation drawer)
- Botones, botones de imagen en la pantalla principal
- Otras vistas en las que se puede hacer clic con texto e iconos dispuestos en filas horizontales o verticales, o como una cuadrícula
- Listar elementos en las pantallas de colección.

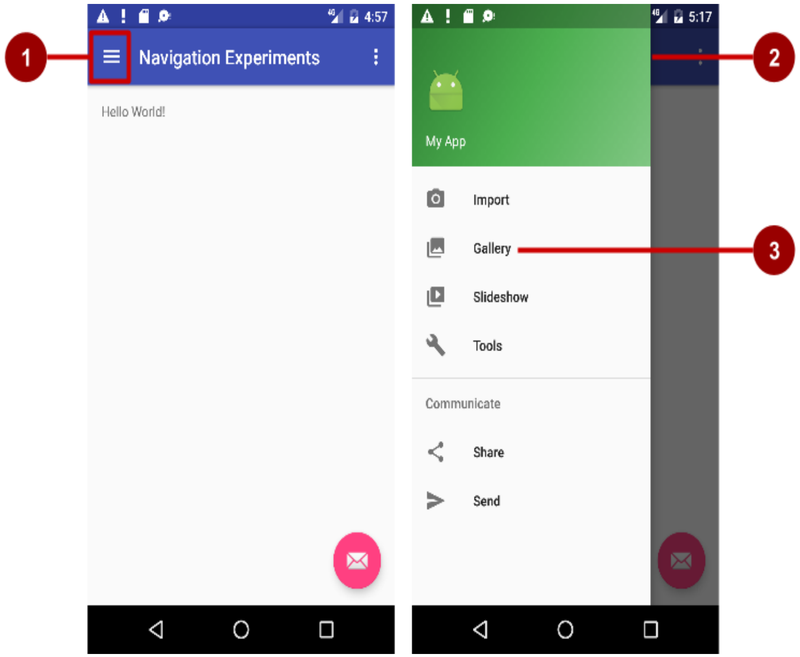
MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Navigation drawer

Navegación descendiente
1. Icono en la barra de aplicaciones
2. Encabezamiento
3. Elementos de menú



MODULOS

Las pantallas en un proyecto Android

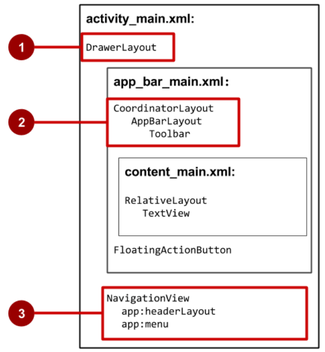
Navegación de usuario; Dos formas de navegación

Crea diseños:

- Un navigation drawer como Activity layout root ViewGroup
- Una vista de navegación para el propio drawer
- Un diseño de barra de aplicaciones que incluye espacio para un botón de icono de navegación
- Un diseño de contenido para la actividad que muestra el panel de navegación.
- Un diseño para el encabezado del drawer de navegación

Diseño de actividad del cajón de navegación

1. es la vista raíz
2. CoordinatorLayout contiene el diseño de la barra de aplicaciones con una barra de herramientas
3. Diseño de la pantalla de contenido de la aplicación
4. NavigationView con diseños para encabezado y elementos seleccionables



MODULOS

Las pantallas en un proyecto Android

Navegación de usuario; Dos formas de navegación

Pasos para implementar el cajón de navegación

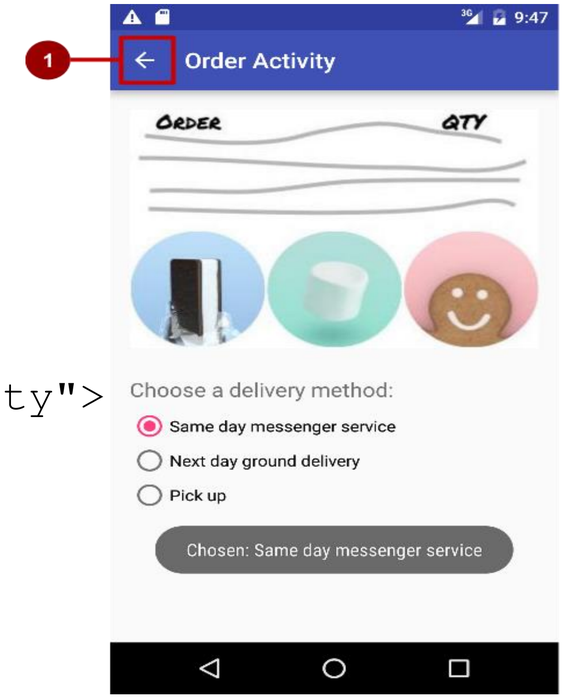
- Complete el menú del cajón de navegación con títulos e íconos de elementos
- Configure el cajón de navegación y los oyentes de elementos en el código de actividad
- Manejar las selecciones de elementos del menú de navegación
- Complete el menú del cajón de navegación con títulos e íconos de elementos
- Configurar el cajón de navegación y los oyentes de elementos en el código de actividad
- Manejar las selecciones de elementos del menú de navegación
- Lista vertical, como RecyclerView
- Cuadrícula vertical, como GridView
- Navegación lateral con carrusel
- Menús de varios niveles, como el menú de opciones
- Flujo de navegación maestro / detallado

MODULOS

Las pantallas en un proyecto Android: Fragment

Navegación ancestral (botón Arriba)

Permitir que el usuario suba de una sección o pantalla secundaria a la pantalla principal



Como declarar parent of child Activity en AndroidManifest.xml

```
<activity android:name=".OrderActivity"
  android:label="@string/title_activity_order"
  android:parentActivityName="com.example.android.
                    optionsmenuorderactivity.MainActivity">
  <meta-data
     android:name="android.support.PARENT_ACTIVITY"
     android:value=".MainActivity"/>
</activity>
```

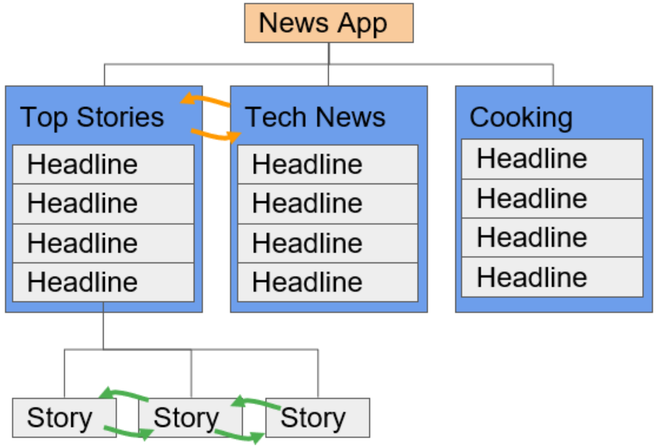
MODULOS

Las pantallas en un proyecto Android: Fragment

Navegación lateral

Pestañas y deslizamientos:

Navegación lateral
Entre paginas hermanas
De una lista de historias a una lista en una pestaña diferente
De historia en historia en la misma pestaña



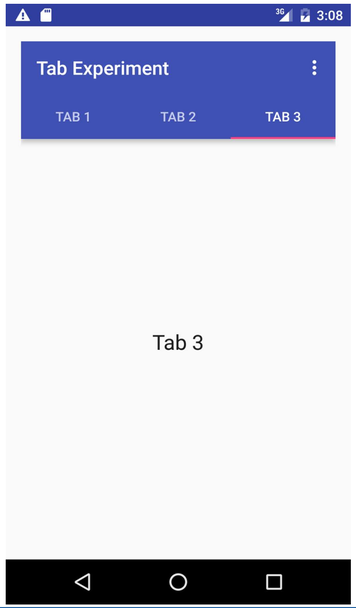
MODULOS

Las pantallas en un proyecto Android: Fragment

Navegación lateral

Entre los beneficios de usar pestañas y deslizamientos se puede mencionar que una sola pestaña seleccionada inicialmente permite que los usuarios tengan acceso al contenido sin tener que navegar más y navegar entre pantallas relacionadas sin visitar a las paginas padres

Mejores prácticas con pestañas
Disposición horizontal
Corre por la parte superior de la pantalla
Persistente en pantallas relacionadas
El cambio no debe tratarse como un historial



MODULOS

Las pantallas en un proyecto Android: Fragment

Navegación lateral, Pasos para implementar pestañas:

- Definir el diseño de la pestaña usando TabLayout
- Implementar un Fragmento y su diseño para cada pestaña
- Implementar un PagerAdapter de FragmentPagerAdapter o FragmentStatePagerAdapter
- Crea una instancia del diseño de pestaña
- Utilice PagerAdapter para administrar pantallas (cada pantalla es un fragmento)
- Establecer un oyente para determinar qué pestaña se toca

```
<android.support.design.widget.TabLayout
    android:id="@+id/tab_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/toolbar"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>
```

MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView. ¿Qué es Recycler View?

RecyclerView es un contenedor desplazable para grandes conjuntos de datos

Es Eficiente

Utiliza y reutiliza un número limitado de elementos de la vista
Actualiza los datos cambiantes rápidamente

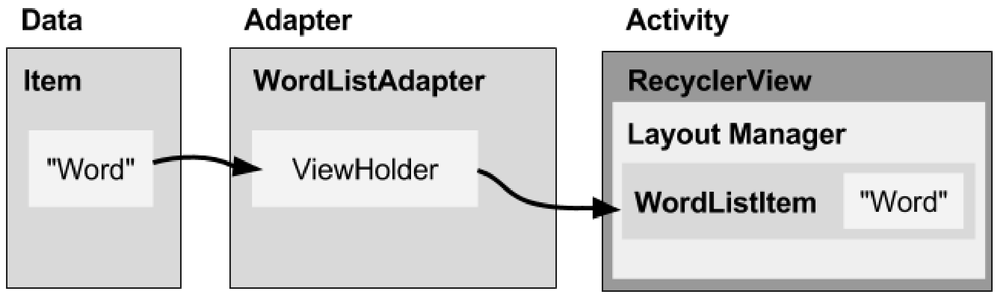
MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView. ¿Cómo todos sus componentes trabajan en conjunto?

Datos

- Lista de desplazamiento de RecyclerView para elementos de lista: RecyclerView
- Diseño para un elemento de datos: archivo XML
- El administrador de diseño maneja la organización de los componentes de la interfaz de usuario en una vista: Recyclerview.
- El adaptador conecta datos a RecyclerView — RecyclerView.Adapter
- ViewHolder tiene información de vista para mostrar un elemento: RecyclerView.ViewHolder



MODULOS

Las pantallas en un proyecto Android: Layout.xml

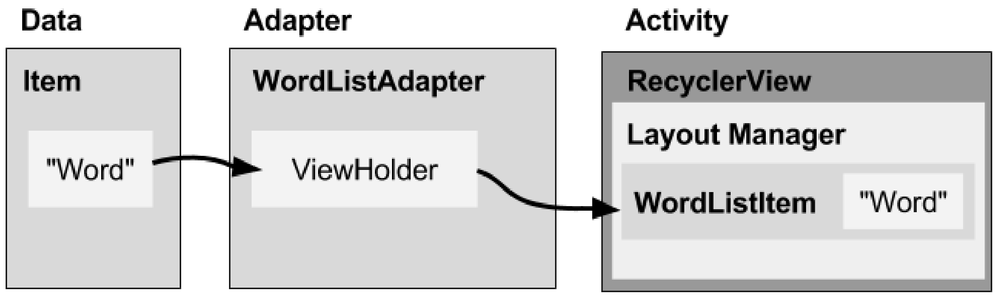
RecyclerView. ¿Cómo todos sus componentes trabajan en conjunto?

Cada ViewGroup tiene un administrador de diseño
Úselo para colocar elementos de Vista dentro de un RecyclerView
Reutiliza elementos de la vista que ya no son visibles para el usuario

Administradores de diseño integrados:

- LinearLayoutManager
- GridLayoutManager
- StaggeredGridLayoutManager

Hace “extend” a RecyclerView.LayoutManager



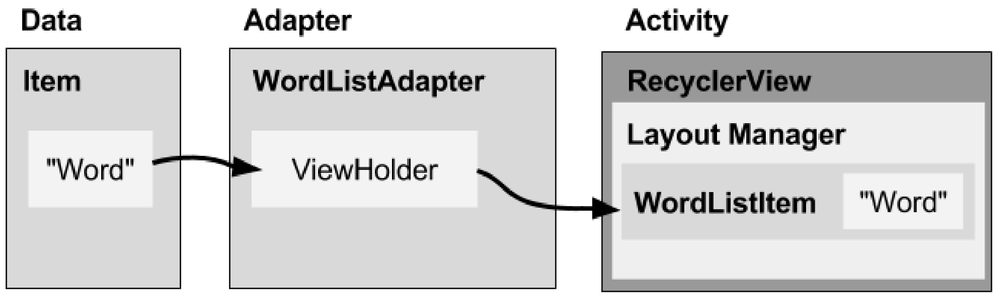
MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView. ¿Qué es un Adapter?

Ayuda a que las interfaces incompatibles funcionen juntas
Ejemplo: toma datos del cursor de la base de datos y prepara cadenas para poner en una vista
Intermediario entre datos y vista

Gestiona la creación, actualización, adición y eliminación de elementos de visualización como cambios de datos subyacentes via:
RecyclerView.Adapter

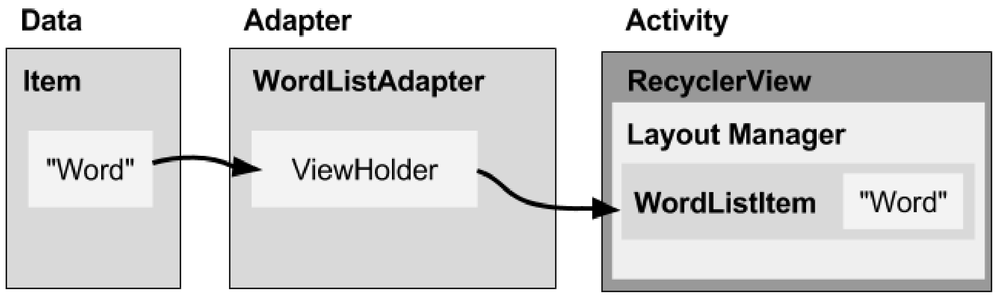


MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView. ¿Cómo es el paso a paso de su implementación?

Agregue la dependencia RecyclerView a build.gradle si es necesario
Agregar RecyclerView al diseño
Crear diseño XML para el artículo
Amplíe RecyclerView.Adapter
Amplíe RecyclerView.ViewHolder
En Activity onCreate (), cree RecyclerView con el adaptador y el administrador de diseño



MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView.

XML layout

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</android.support.v7.widget.RecyclerView>
```

MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView.

Ejemplo de layout con lista de 1 item

```
<LinearLayout …>
  <TextView
    android:id="@+id/word"
    style="@style/word_title" />
</LinearLayout>
```

MODULOS

Las pantallas en un proyecto Android: Layout.xml

RecyclerView.

Implementación de adaptador

```
public class WordListAdapter
    extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {

    public WordListAdapter(Context context,
                           LinkedList<String> wordList) {
        mInflater = LayoutInflater.from(context);
        this.mWordList = wordList;
    }
}
```

MODULOS

El editor de vistas

El adaptador tiene tres métodos requeridos:

- onCreateViewHolder()

```
@Override
public WordViewHolder onCreateViewHolder(
        ViewGroup parent, int viewType) {
        // Create view from layout
        View mItemView = mInflater.inflate(
             R.layout.wordlist_item, parent, false);
        return new WordViewHolder(mItemView, this);
}
```

MODULOS

Las pantallas en un proyecto Android: El editor de vistas

El adaptador tiene tres métodos requeridos:

- getItemCount()

```
@Override
 public int getItemCount() {
   // Return the number of data items to display
   return mWordList.size();
 }
```

MODULOS

Las pantallas en un proyecto Android: El editor de vistas

El adaptador tiene tres métodos requeridos:

- onBindViewHolder()

```
@Override
public void onBindViewHolder(
        WordViewHolder holder, int position) {
        // Retrieve the data for that position
        String mCurrent = mWordList.get(position);
        // Add the data to the view
        holder.wordItemView.setText(mCurrent);
}
```


Posiciona