Trabajo diseño Android

María José Murcia Martínez

SENA – Servicio Nacional de Aprendizaje

Centro de la Industria la Empresa y los Servicios

Análisis y Desarrollo de Software

Ficha: 2694667

Carlos Julio Cadena

27 de mayo del 2024

Neiva / Huila

1. De los siguientes contenedores:
   * + **LinearLayout (vertical y horizontal):**

- Características principales:

* Organiza los elementos de forma lineal, ya sea vertical u horizontalmente.
* Permite definir el peso de los elementos para distribuir el espacio de manera proporcional.
* Simple y fácil de usar.

- Como se organizan los objetos dentro del contenedor:

* Vertical: Los elementos se organizan en una columna, uno debajo del otro.
* Horizontal: Los elementos se organizan en una fila, uno al lado del otro.

- Ventajas:

* Fácil de entender y usar.
* Predecible en su comportamiento.

- Desventajas:

* Puede resultar en layouts ineficientes si se usan muchos elementos anidados, afectando el rendimiento.
* Menos flexible comparado con otros contenedores como ConstraintLayout.

- Ejemplo de interfaz:



* + - **RelativeLayout:**

- Características principales:

* Permite posicionar los elementos relativos entre sí o al contenedor principal.
* Proporciona más flexibilidad en la colocación de los elementos.

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan en relación con otros objetos o al contenedor padre, utilizando layout\_below, layouts\_above, layout\_toLeftOf, etc.

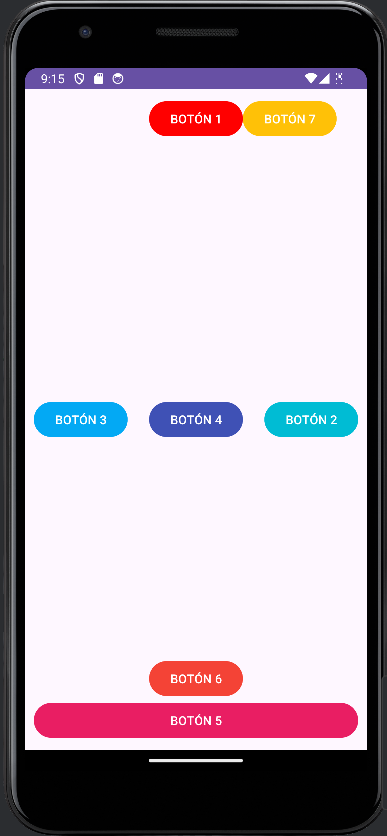
- Ventajas:

* Muy flexible para crear layouts complejos.
* Reduce la necesidad de anidación excesiva de contenedores.

-Desventajas:

* El rendimiento puede verse afectado si no se usa correctamente.
* Puede ser complicado y difícil de mantener con layouts complejos.

- Ejemplo de interfaz:

****

* + - **FrameLayout:**

- Características principales:

* Diseñado para contener un solo hijo, aunque puede contener múltiples.
* Los elementos se apilan uno sobre otro, con el último agregado visible en la parte superior.

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan en la esquina superior izquierda de la pantalla, apilados unos sobre otros.

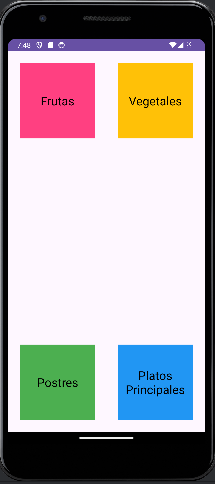
- Ventajas:

* Ideal para layouts simples con una sola vista o para usar como un contenedor de fondo.
* Eficiente en términos de rendimiento.

-Desventajas:

* No es adecuado para layouts complejos.
* Difícil de manejar si se necesita controlar la superposición de múltiples vistas.

- Ejemplo de interfaz:

****

* + - **ConstraintLayout:**

- Características principales:

* Permite crear layouts complejos sin anidación profunda de vistas.
* Los elementos se posicionan utilizando restricciones en lugar de anidación.

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan mediante la creación de restricciones en sus posiciones relativas a otros elementos y al contenedor padre.

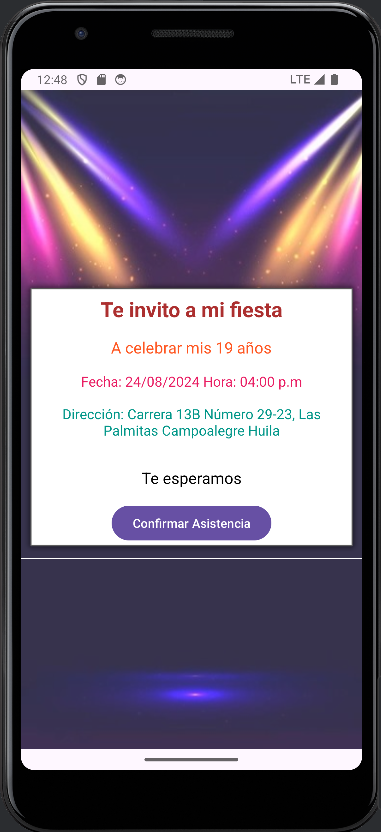
- Ventajas:

* Reduce la profundidad de la jerarquía de vistas.
* Muy flexible y potente para crear interfaces complejas.

-Desventajas:

* Requiere una curva de aprendizaje.
* Puede ser excesivo para layouts muy simples.

- Ejemplo de interfaz:

****

* + - **CardView:**

- Características principales:

* Extiende FrameLayout y ofrece un contenedor con bordes redondeados y sombra.
* Comúnmente usado para mostrar elementos de listas con estilo de tarjeta.

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan dentro de un FrameLayout con bordes y sombra aplicados.

- Ventajas:

* Atractivo visualmente con sombras y bordes redondeados.
* Simple de usar y configurar.

-Desventajas:

* Añade algo de complejidad y carga al rendimiento debido a las sombras y bordes.

- Ejemplo de interfaz:

**Texto

Descripción generada automáticamente con confianza media**

* + - **RecyclerView:**

- Características principales:

* Más flexible y eficiente que ListView para mostrar grandes cantidades de datos.
* Utiliza un adaptador para gestionar los datos y las vistas.
* Soporta layout managers para definir la disposición de los elementos (vertical, horizontal, grid).

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan según el LayoutManager elegido (LinearLayoutManager, GridLayoutManager, etc.).

- Ventajas:

* Altamente flexible y personalizable.
* Mejor rendimiento debido al uso de ViewHolder pattern.

-Desventajas:

* Requiere más código y configuración comparado con ListView.
* Mayor complejidad en su implementación.

- Ejemplo de interfaz:

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

* + - **ListView:**

- Características principales:

* Muestra una lista de elementos desplazables verticalmente.
* Utiliza un adaptador para enlazar los datos a las vistas.

- Como se organizan los objetos dentro del contenedor:

* Los objetos se organizan en una lista vertical, cada elemento es una vista definida por el adaptador.

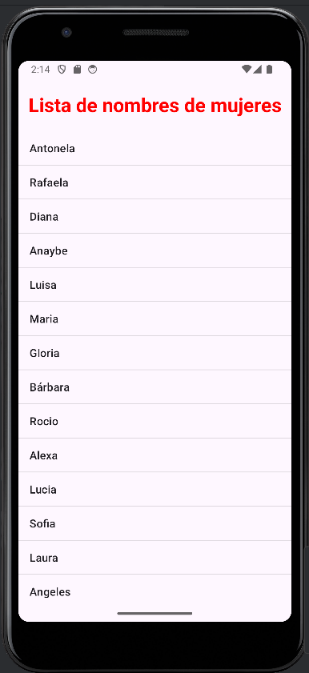
- Ventajas:

* Fácil de implementar y usar para listas simples.
* Buen rendimiento para listas pequeñas a moderadas.

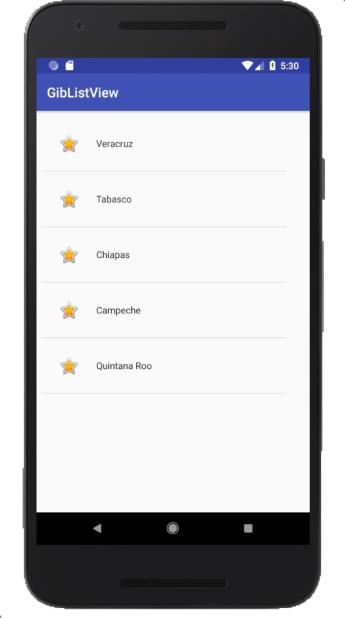
-Desventajas:

* Menos flexible y personalizable que RecyclerView.
* Menor rendimiento con listas muy grandes debido a la falta de ViewHolder pattern.

- Ejemplo de interfaz:



1. Diseñar una interfaz con estas características:



  Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente