TALLER ANDROID

**INTEGRANTE:**

ANYI ZUJEY GOMEZ CASANOVA

**INSTRUCTOR:**

CARLOS JULIO CADENA

**TECNOLOGO:**

ANALISIS Y DESARROLLO DE SOFTWARE

SERVICIO NACIONAL DE APRENDIZAJE **SENA**

NEIVA – HUILA

2024

1.De los siguientes contenedores:

•LinearLayout(vertical y horizontal).

•RelativeLayout.

•FrameLayout.

•ConstraintLayout

•CardView

•RecyclerView

•ListView

-Características principales.

-Como se organizan los objetos dentro del contenedor

-Ventajas y desventajas.

-Ejemplo de interfaz.

**LinearLayout(vertical y horizontal):**

Características principales:

* Organiza los elementos en una sola dirección: vertical u horizontal.
* Es fácil de usar y entender.

Organización de objetos: Los objetos se colocan uno después del otro en el orden en que se añaden, ya sea verticalmente o horizontalmente.

Ventajas:

* Es simple y fácil de implementar.
* Es útil para organizar elementos en una sola fila o columna.

Desventajas:

* No es muy eficiente para diseños complejos.

Ejemplo de interfaz: Una pantalla de inicio con botones dispuestos verticalmente para diferentes funciones, como "Inicio", "Configuración" y "Salir".

**RelativeLayout.:**

Características principales:

* Permite posicionar los elementos relativamente entre sí o con respecto al contenedor padre.
* Es útil para diseños más complejos y flexibles.

Organización de objetos: Los objetos se posicionan en relación con otros elementos secundarios o con el contenedor padre.

Ventajas:

* Flexibilidad en el posicionamiento de elementos.
* Útil para diseños más complejos y personalizados.

Desventajas:

* Puede ser más complicado de entender y mantener en diseños muy complejos.

Ejemplo de interfaz: Un botón alineado a la parte inferior derecha de la pantalla independientemente del tamaño de la pantalla.

**FrameLayout.:**

Características principales:

* Muestra un único elemento hijo a la vez, superponiendo cada nuevo elemento hijo sobre el anterior.
* Útil para la superposición de elementos.

Organización de objetos: Los objetos se superponen uno sobre otro en el orden en que se añaden.

Ventajas:

* Simple y ligero.
* Ideal para superposición de elementos como fragmentos o vistas.

Desventajas:

* Limitado a mostrar un solo elemento hijo a la vez.

Ejemplo de interfaz: Un ImageView superpuesto con un TextView para mostrar una imagen con texto sobre ella.

**ConstraintLayout:**

Características principales:

* Permite definir relaciones entre elementos secundarios para crear diseños complejos y adaptables.
* Es altamente flexible y eficiente en cuanto a recursos.

Organización de objetos: Los objetos se colocan y dimensionan en función de las restricciones definidas entre ellos y el contenedor padre.

Ventajas:

* Alta flexibilidad y eficiencia.
* Perfecto para diseños adaptables y complejos.

Desventajas:

* Puede tener una curva de aprendizaje más empinada debido a su flexibilidad.

Ejemplo de interfaz: Una pantalla de inicio con elementos que se ajustan automáticamente a diferentes tamaños de pantalla y orientaciones.

**CardView:**

Características principales:

* Muestra contenido en una tarjeta elevada con sombra para una presentación visualmente atractiva.
* Ideal para presentar información en forma de tarjetas independientes.

Organización de objetos: El contenido se coloca dentro del CardView como un solo elemento.

Ventajas:

* Añade profundidad visual a la interfaz de usuario.
* Útil para mostrar información de manera clara y organizada.

Desventajas:

* Puede aumentar la complejidad visual si se usa en exceso.

Ejemplo de interfaz: Una lista de perfiles de usuario donde cada perfil está contenido en un CardView con la imagen del usuario, su nombre y una breve descripción.

**RecyclerView:**

Características principales:

* Muestra una colección de elementos en una lista desplazable de manera eficiente.
* Ideal para manejar grandes conjuntos de datos.

Organización de objetos: Los elementos se presentan en una lista desplazable.

Ventajas:

* Eficiente para manejar grandes cantidades de datos.
* Permite el reciclaje de vistas para un mejor rendimiento.

Desventajas:

* Requiere más configuración que ListView.

Ejemplo de interfaz: Una lista de mensajes en una aplicación de chat, donde cada mensaje se muestra como un elemento individual en la lista desplazable.

ListView:

Características principales:

* Muestra una lista desplazable de elementos.
* Aunque ha sido reemplazado por RecyclerView en la mayoría de los casos, todavía se encuentra en algunas aplicaciones existentes.

Organización de objetos: Los elementos se presentan en una lista desplazable.

Ventajas:

* Fácil de implementar para listas simples.
* Funciona bien para pequeñas cantidades de datos.

Desventajas:

* Menos eficiente que RecyclerView para grandes conjuntos de datos.

Ejemplo de interfaz: Una lista de contactos en una aplicación de teléfono, donde cada contacto se muestra como un elemento en la lista desplazable.