

3 - Listas estáticas de datos (LazyColumn y LazyRow)

Cuando se necesita mostrar colecciones grandes de datos que requieren el scroll de pantalla, ya sea vertical u horizontal, disponemos de las funciones componibles LazyColumn y LazyRow.

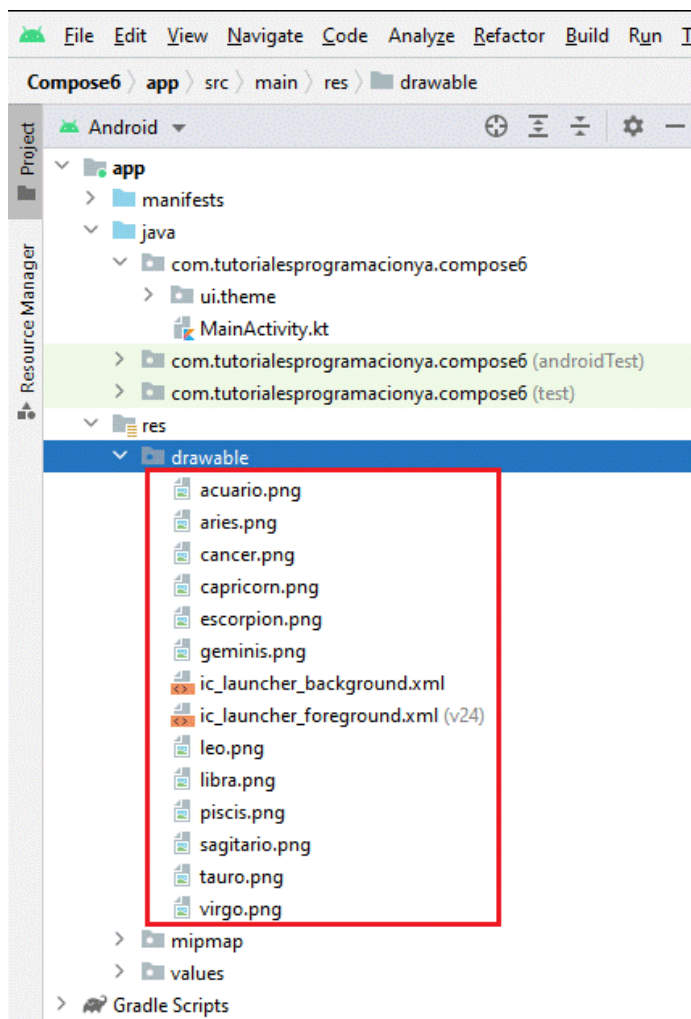
Estas funciones son muy eficientes para mostrar gran cantidad de datos, que no pueden ser tratadas por las funciones que vimos anteriormente: Column y Row.

LazyColumn genera una lista de desplazamiento vertical, mientras que LazyRow produce una de desplazamiento horizontal.

Problema

Mostrar los signos de Zodíaco, sus fechas y una descripción de cada uno mediante una lista.

1. Creamos el proyecto 'Compose6'
2. Copiamos a la carpeta drawable las 12 imágenes de los signos del zodiaco:



Ahora codificamos:

```
package com.teji.compose6

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            ListadoSignos(listaDeSignos = listaDeSignos)
        }
    }
}
```

```
@Composable
fun ListadoSignos(listaDeSignos: List<Signo>) {
    LazyColumn {
        items(listaDeSignos) { signoactual ->
            TarjetaSigno(signo = signoactual)
        }
    }
}

@Composable
fun TarjetaSigno(signo: Signo) {
    Text(text = signo.nombre,
        style = MaterialTheme.typography.h3)
    Image(
        painter = painterResource(id = signo.imagen),
        contentDescription = null,
        contentScale = ContentScale.Crop,
        modifier = Modifier.fillMaxSize()
    )
    Text(text = signo.titulo)
    Text(
        text = signo.descripcion,
        style = MaterialTheme.typography.body1
    )
}
```

```
data class Signo(val nombre: String, val titulo: String, val descripcion: String, val imagen: Int)
```

```
val listaDeSignos = listOf(  
    Signo(  
        "Aries",  
        "Aries (21 de marzo-20 de abril)",  
        "Los nacidos bajo el signo de Aries suelen ser personas muy adaptables, que pueden llevarse bien con los demás por su creatividad y espontaneidad. Son detallistas y observadores.",  
        R.drawable.aries  
    ),  
    Signo(  
        "Tauro",  
        "Tauro (21 de abril- 21 de mayo)",  
        "Este es un signo amoroso, al que le gusta ser romántico en sus mejores momentos. También es amante de la fuerza y de la resistencia, con una gran voluntad para hacer las cosas y encaminarlas hacia el camino correcto.",  
        R.drawable.tauro  
    ),  
    Signo(  
        "Géminis",  
        "Géminis (22 de mayo-21 de junio)",  
        "La inteligencia es una de sus cualidades más conocidas y reconocidas, pero no sólo de mente
```

, también a la hora de actuar. Suelen ser personas equilibradas, adaptables y muy entregadas al amor."
,

R.drawable.geminis

),

Signo(

"Cáncer",

"Cáncer (22 de junio-22 de julio)",

"Son personas hogareñas, románticas y muy apasionadas. Les encanta compartir en familia todos los momentos, además de ser entregadas y dedicadas. Suelen ser personas admiradas y muy simpáticas, por lo que le suelen caer bien a todo el mundo."

,

R.drawable.cancer

),

Signo(

"Leo",

"Leo (23 de julio-23 de agosto)",

"Les gusta ser líderes, apoyándose en sus ideas y convicciones. Les encanta llamar la atención y tener la de los demás siempre sobre de ellos, aunque hay veces que les gana la soberbia."

R.drawable.leo

),

Signo(

"Virgo",

"Virgo (24 de agosto-23 de septiembre)",

"Son personas de carácter fuerte, con ideas firmes y claras, pues cuando quieren algo lo consiguen a como dé lugar. Tienen habilidad para conv

encer a los demás, para ser el centro de atracción y el alma de las fiestas o las reuniones.",

R.drawable.virgo

),

Signo(

"Libra",

"Libra (24 de septiembre-23 de octubre)",

"El equilibrio siempre está presente en su vida, al ser amante de la estabilidad y la paridad en todos los sentidos. Es gente tranquila, a la que le gusta la armonía y hasta la soledad en algunos momentos, aunque también puede resultar todo lo contrario, con desorden y ruido.",

R.drawable.libra

),

Signo(

"Escorpión",

"Escorpión (24 de octubre-22 de noviembre)

",

"De mente calculadora, con carácter fuerte, pero bondadoso en el fondo. Hábiles para negociar o alcanzar sus metas. Los obstáculos se convierten en retos que por lo regular superan sin miramientos. También son apasionados.",

R.drawable.escorpion

),

Signo(

"Sagitario",

"Sagitario (23 de noviembre-21 de diciembre)",

"Suelen ser desordenados, atrabancados o hasta berrinchudos si las cosas no se dan como las pensaron. En contraparte, son capaces de enfocar toda su energía para solucionar dificultades.",

R.drawable.sagitario

),

Signo(

"Capricornio",

"Capricornio (22 de diciembre-20 de enero)

",

"Prácticas, con una habilidad nata para encontrarle solución a las cosas, aunque parezcan casos perdidos. Son amantes del orden, la estabilidad y de que todo camine conforme lo han planeado.",

R.drawable.capricorn

),

Signo(

"Acuario",

"Acuario (21 de enero-18 de febrero)",

"Amorosas, cariñosas y muy sensibles suelen ser las personas nacidas bajo este signo. Les gustan las causas nobles y el dar sin esperar algo a cambio.",

R.drawable.acuario

),

Signo(

"Piscis",

"Piscis (19 de febrero-20 de marzo)",

"La honestidad puede llegar a ser una de sus mejores cualidades. Manejan un carácter tranquilo, muy aliviado y consolador. Creen en los demás

```
s, aunque eso les pueda acarrear alguna dificultad
al ser mal pagados.",
        R.drawable.piscis
    )
)
```

Antes de analizar la parte visual, veamos donde están almacenados los datos que se deben mostrar. Declaramos un data class llamado Signo :

```
data class Signo(val nombre: String, val titulo: String, val descripcion: String, val imagen: Int)
```

Llamamos a la función listOf que nos devuelve un objeto de la clase List y almacenamos 12 objetos de la clase Signo:

```
val listaDeSignos = listOf(
```

```
    Signo(
```

```
        "Aries",
```

```
        "Aries (21 de marzo-20 de abril)",
```

```
        "Los nacidos bajo el signo de Aries suelen ser personas muy adaptables, que pueden llevarse bien con los demás por su creatividad y espontaneidad. Son detallistas y observadores.",
```

```
        R.drawable.aries
```

```
    ), ..... y los otros 11 objetos
```

Es decir listaSignos es un objeto de la clase List con 12 elementos de tipo Signo.

Ahora veamos las dos funciones componibles, la primera la llamamos desde el método onCreate del Activity y le pasamos la lista a mostrar:

```
class MainActivity : ComponentActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContent {
```

```
            ListadoSignos(listaDeSignos = listaDeSignos)
```

```
        }
```

```
    }
```

```
}
```


La función componible ListadoSignos recibe como parámetro la lista. Llamamos a la función LazyColumn, en su interior mediante la función items iteramos por cada elemento de la listaDeSignos y para cada uno de los signos llamamos a la segunda función componible TarjetaSigno a la cual le enviamos en forma sucesiva cada uno de los signos almacenados en la listaDeSignos:

@Composable

```
fun ListadoSignos(listaDeSignos: List<Signo>) {  
    LazyColumn {  
        items(listaDeSignos) { signoactual ->  
            TarjetaSigno(signo = signoactual)  
        }  
    }  
}
```

La segunda función composible es TarjetaSigno que recibe como parámetro un objeto de tipo Signo. En su interior mostramos los datos del signo que llega como parámetro, debemos mostrar el nombre del signo en el primer Text.

Para mostrar la imagen procedemos a llamar a la función composible Image, indicando en el parámetro painter la referencia de la imagen.

De forma similar mostramos el título y la descripción en otras dos llamadas a las funciones composibles Text:

@Composable

```
fun TarjetaSigno(signo: Signo) {  
    Text(text = signo.nombre,  
        style = MaterialTheme.typography.h3)  
    Image(  
        painter = painterResource(id = signo.imagen),  
        contentDescription = null,  
        contentScale = ContentScale.Crop,  
        modifier = Modifier.fillMaxSize()  
    )  
    Text(text = signo.titulo)  
    Text(  
        text = signo.descripcion,  
        style = MaterialTheme.typography.bodyText2,  
        modifier = Modifier.fillMaxWidth()  
    )  
}
```

```

        text = signo.descripcion,
        style = MaterialTheme.typography.body1
    )
}

```

Con este ejemplo podemos verificar que es muy sencillo implementar pantallas con listas de datos estáticos con scroll:

