

# Apuntes

## Tema 6 Listas en Compose

Descargar estos apuntes [pdf](#) o [html](#)

Índice

- [Introducción.](#)
- [LazyColumn.](#)
-

# Introducción.

Las listas son una herramienta muy útil en el desarrollo de aplicaciones porque permiten mostrar una gran cantidad de elementos de manera ordenada y facilitando el modo de navegar para el usuario, presentando beneficios en términos de usabilidad y experiencia de usuario.

Las listas también permiten a los usuarios realizar acciones en masa, como seleccionar varios elementos y realizar una acción en todos ellos al mismo tiempo. Por ejemplo, en una aplicación de correo electrónico, los usuarios pueden seleccionar varios correos electrónicos en una lista y luego eliminarlos todos juntos.

Si necesitamos mostrar una gran cantidad de elementos (o una lista de longitud desconocida), el uso de `Column` puede no ser eficiente. *Compose* proporciona un conjunto de componentes para presentar listas: `LazyColumn`, `LazyRow` y `LazyVerticalGrid`.

Veamos estos componentes uno a uno.

## LazyColumn.

Creemos un proyecto nuevo que se llame `iMediaItem`.

El aspecto de nuestra aplicación sería el siguiente:



Y la estructura de carpetas asociada a nuestra arquitectura sería la siguiente:



En la carpeta `\models` encontramos el archivo *MediaItem.kt*, siendo esta `data class` nuestro *POJO*:

```
data class MediaItem(
    val id: Int,
    val title: String,
    val thumb: String,
    val type: Type
) {
    enum class Type { PHOTO, VIDEO }
}
```

En la carpeta `\data` almanamos la información referente a nuestros datos, como no sabemos todavía cómo acceder a *BD* o a otras fuentes de información, utilizaremos un proveedor propio.

```
//La función getMedia genera 20 elementos de tipo MediaItem, uno de cada tres es un video
class MediaItemProvider {
    companion object {
        private val mediaItemList = (1..20).map {
            MediaItem(
                id = it,
                title = "Title $it",
                thumb = "https://lorempixel.com/400/400/people/$it/",
                type = if (it % 3 == 0) MediaItem.Type.VIDEO else MediaItem.Type.PHOTO
            )
        }

        fun getMedia() = mediaItemList
    }
}
```

La definición de un `LazyColumn` es la siguiente:

```
@Composable
fun LazyColumn(
    modifier: Modifier! = Modifier,
    state: LazyListState! = rememberLazyListState(),
    contentPadding: PaddingValues! = PaddingValues(0.dp),
    reverseLayout: Boolean! = false,
    verticalArrangement: Arrangement.Vertical! = if (!reverseLayout) Arrangement.Top else Arr
    horizontalAlignment: Alignment.Horizontal! = Alignment.Start,
    flingBehavior: FlingBehavior! = ScrollableDefaults.flingBehavior(),
    userScrollEnabled: Boolean! = true,
    content: (@ExtensionFunctionType LazyListScope.() -> Unit)?
): Unit
```



