

# LISTAS CON LAZY COMPOSABLE

# LISTAS CON **LAZY COMPOSABLE**

- En lecciones anteriores vimos cómo implementar listas de elementos a través de los componentes **Column** y **Row**.
- Cuando el número de elementos a mostrar es grande, es preferible usar componentes Lazy (**LazyColumn** o **LazyRow**):
  - Implementación de scroll de forma automática.
  - Reciclaje de elementos de la lista.
  - Mismos principios que el componente **RecyclerView**.

# LISTAS CON **LAZY COMPOSABLE**

- La diferencia entre **LazyColumn** y **LazyRow** es la orientación en la que se integran sus elementos y se desplazan.
- **LazyColumn** produce un desplazamiento vertical mientras que **LazyRow** produce un desplazamiento horizontal.

# LISTAS CON **LAZY COMPOSABLE**

## LazyListScope:

- Al igual que **Column** y **Row**, los componentes Lazy Composable ofrecen un **Scope** para añadir contenido.
- En el caso de **LazyListScope**, se ofrece un conjunto de funciones para añadir elementos a la lista.

# LISTAS CON LAZY COMPOSABLE

## LazyListScope:

- **item:** Agrega un solo elemento a la lista.
- **items(N):** Agrega varios elementos a la lista

```
@Composable
fun ItemList() {
    LazyColumn { this: LazyListScope
        // Add a single item
        item { this: LazyItemScope
            Text(text = "First item")
        }

        // Add 3 items
        items(count: 3) { index ->
            Text(text = "Item: $index")
        }

        // Add another single item
        item { this: LazyItemScope
            Text(text = "Last item")
        }
    }
}
```



# LISTAS CON LAZY COMPOSABLE

## LazyListScope:

- Existen funciones de extensión que permiten agregar colecciones de elementos como **List**.

```
@Composable
fun MessageList(messages: List<String>) {
    LazyColumn { this: LazyListScope
        items(messages) { message ->
            MessageRow(message)
        }
    }
}

@Composable
fun MessageRow(message: String) {
    Text(text = message)
}
```

# LISTAS CON LAZY COMPOSABLE

## PaddingValues:

- LazyColumn permite añadir parámetros del tipo **PaddingValues** al parámetro **contentPadding**.
- **16.dp** de padding a los bordes horizontales (izquierda y derecha) y **8.dp** al principio y al final del contenido.

```
LazyColumn(  
    contentPadding = PaddingValues(  
        horizontal = 16.dp,  
        vertical = 8.dp  
    )  
) {  
    this: LazyListScope  
    // ...  
}
```

# LISTAS CON LAZY COMPOSABLE

## Arrangement:

- Para agregar espaciado entre elementos, puede usarse **Arrangement.spacedBy**.
- En el siguiente ejemplo, se agregan 4.dp de espacio entre cada elemento:

```
LazyColumn(  
    verticalArrangement =  
        Arrangement.spacedBy(4.dp)  
) {  
    this: LazyListScope  
    // ...  
}
```