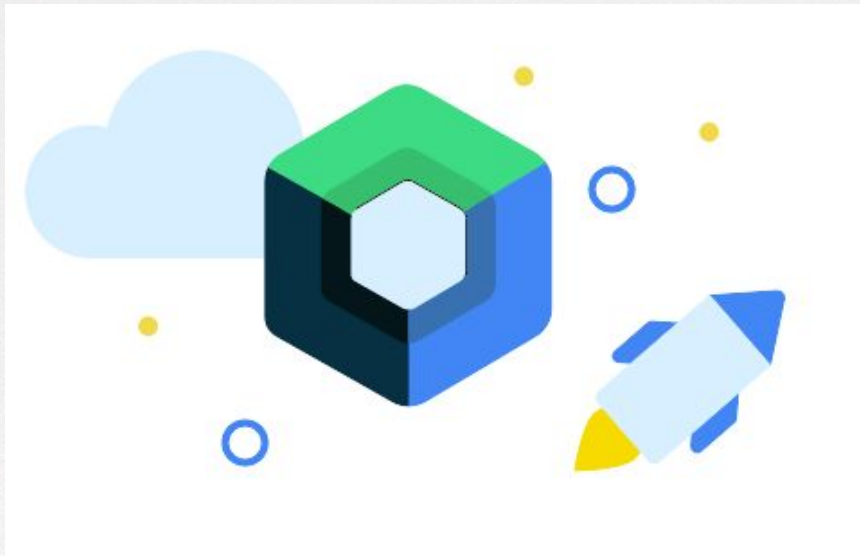


# ¿QUÉ ES JETPACK COMPOSE?

# ¿QUÉ ES JETPACK COMPOSE?



- Nueva forma de programar interfaces en Android.
- Simplifica la creación de vistas.
- Es más intuitivo.
- Acelera el desarrollo.
- Muy potente.



**Jetpack Compose se convertirá este año 2021 en el nuevo estándar de desarrollo de interfaces de usuario en Android.**

# ¿QUÉ ES **JETPACK COMPOSE**?



- Google ha cerrado la definición del API.
- Versión estable 28 julio 2021.
- Desarrollado con feedback continuo de la comunidad.
- Se calcula que ya hay unas 2000 apps en Play Store que usan Jetpack Compose.

# ¿QUÉ ES **JETPACK COMPOSE**?



Swift UI



Flutter



Jetpack Compose



# ¿QUÉ ES **JETPACK COMPOSE**?

## **Menos código**

Más con menos código. Código más simple y más fácil de mantener.

## **Más intuitivo**

Declara tu interfaz y Compose se encargará del resto

## **Más reutilizable**

Define componentes y reutilízalos con facilidad.

## **Vistas en Kotlin**

Programa tu interfaz en el mismo lenguaje que tu lógica de negocio.

# ¿QUÉ ES JETPACK COMPOSE?

## RecyclerView tradicional

RecyclerViewActivity class

```
class RecyclerViewActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_recycler)  
        recycler_view.apply { this: RecyclerView!  
            layoutManager = LinearLayoutManager( context: this@RecyclerViewActivity)  
            adapter = RecyclerViewAdapter(listOf(Item( name: "test name", description: "test description")))  
        }  
    }  
}
```

# ¿QUÉ ES JETPACK COMPOSE?

activity\_recycler.xml

```
<androidx.recyclerview.widget.RecyclerView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```



# ¿QUÉ ES JETPACK COMPOSE?

RecyclerViewAdapter class

```
class RecyclerViewAdapter(  
    private val listItems: List<Item>  
) : RecyclerView.Adapter<ItemViewHolder>() {  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder =  
        ItemViewHolder.build(parent)  
  
    override fun getItemCount(): Int = listItems.size  
  
    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {  
        holder.bind(listItems[position])  
    }  
}
```

# ¿QUÉ ES JETPACK COMPOSE?

ViewHolder class

```
class ViewHolder private constructor(
    view: View
) : RecyclerView.ViewHolder(view) {

    companion object {
        fun build(parent: ViewGroup) : ViewHolder =
            LayoutInflater.from(parent.context).inflate(R.layout.item_recycler, parent, attachToRoot: false).let {
                ViewHolder(it)
            }
    }

    fun bind(data: Item) {
        with(itemView) { this: View
            item_name.text = data.name
            item_description.text = data.description
        }
    }
}
```

# ¿QUÉ ES JETPACK COMPOSE?

item\_recycler.xml

```
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    card_view:cardCornerRadius="4dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/item_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:textSize="24sp" />

        <TextView
            android:id="@+id/item_description"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:textSize="24sp" />

    </LinearLayout>
</androidx.cardview.widget.CardView>
```

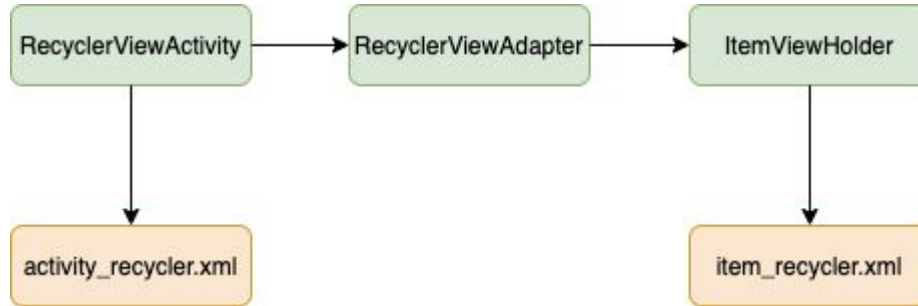
# ¿QUÉ ES JETPACK COMPOSE?

Item class

```
data class Item(  
    val name: String,  
    val description: String  
)
```

# ¿QUÉ ES JETPACK COMPOSE?

RecyclerView class diagram





# ¿QUÉ ES JETPACK COMPOSE?

## RecyclerView con Jetpack Compose

RecyclerViewActivity

```
class RecyclerViewActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            ItemList(  
                itemData = listOf(  
                    Item( name: "test name", description: "test description")  
                )  
            )  
        }  
    }  
}
```

# ¿QUÉ ES JETPACK COMPOSE?

ItemList

```
@Composable
fun ItemList(itemData: List<Item>) {
    LazyColumn { this: LazyListScope
        items(itemData) { itemData ->
            ItemView(item = itemData)
        }
    }
}
```

# ¿QUÉ ES JETPACK COMPOSE?

ItemView

```
@Composable
fun ItemView(item: Item) {
    Card(
        Modifier.fillMaxWidth()
    ) {
        Column { this: ColumnScope
            Text(
                text = item.name,
                fontSize = 24.sp,
                modifier = Modifier.padding(8.dp)
            )
            Text(
                text = item.description,
                fontSize = 24.sp,
                modifier = Modifier.padding(8.dp)
            )
        }
    }
}
```

# ¿QUÉ ES JETPACK COMPOSE?

Item class

```
data class Item(  
    val name: String,  
    val description: String  
)
```

# ¿QUÉ ES JETPACK COMPOSE?

