



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Desarrollo de Aplicaciones Móviles Nativas

“Lista de animes”

Alumno:

Malagón Baeza Alan Adrian

Profesor:

M. en C. José Asunción Enríquez Zárate

Grupo: 7CM1

Introducción

Durante mi práctica, he trabajado en el desarrollo de una aplicación en Android para mostrar una lista de animes junto con sus imágenes. En este informe, presentaré el código relevante y explicaré cómo se implementaron las funcionalidades clave.

El código principal se divide en dos clases: Anime y AnimeAdapter, y una actividad principal llamada MainActivity. La clase Anime representa un anime con su nombre y la referencia a su imagen. La clase AnimeAdapter es un adaptador personalizado que se utiliza para mostrar los elementos de la lista de animes en un RecyclerView. Por último, en la actividad principal (MainActivity), se configura el RecyclerView y se asigna el adaptador para mostrar la lista de animes.

En el adaptador AnimeAdapter, se define una lista de animes con sus nombres y referencias a las imágenes correspondientes. El adaptador se encarga de inflar las vistas de los elementos de la lista y de asignar los valores adecuados a las vistas correspondientes, como el nombre del anime y la imagen.

En la actividad principal (MainActivity), se configura el RecyclerView y se asigna el adaptador. Además, se utiliza un LinearLayoutManager para establecer la orientación de la lista.

Desarrollo

```
class Anime (val name:String, val image: Int)
```

```
class AnimeAdapter: RecyclerView.Adapter<AnimeAdapter.ManejadorVista>() {  
  
    val listaAnimes = listOf<Anime>(  
        Anime("Naruto",R.drawable.naruto),  
        Anime("Attack on Titan",R.drawable.eren),  
        Anime("Demon Slayer",R.drawable.tanjiro),  
        Anime("Sword Art Online",R.drawable.kirito),  
        Anime("Sailor Moon",R.drawable.sailormoon))  
  
    class ManejadorVista(elementoVista: View):  
        RecyclerView.ViewHolder(elementoVista){  
            var imagenAnime: ImageView
```

```

        var nombreAnime: TextView
        init{
            imagenAnime = elementoVista.findViewById(R.id.imagenAnime)
            nombreAnime = elementoVista.findViewById(R.id.nombreAnime)
        }

        override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        ManejadorVista {
            val vista = LayoutInflater.from(parent.context)
                .inflate(R.layout.card_item_layout, parent, false)
            return ManejadorVista(vista)
        }

        override fun getItemCount(): Int {
            // crear un arreglo de elementos a mostrar
            // return elArreglo size
            return listaAnimes.size
        }

        override fun onBindViewHolder(holder: ManejadorVista, position: Int)
        {
            val anime = listaAnimes[position] // Get the Anime object at the
            given position
            holder.imagenAnime.setImageResource(anime.image) // Set the image
            resource
            holder.nombreAnime.text = anime.name // Set the name text
        }
    }
}

```

```

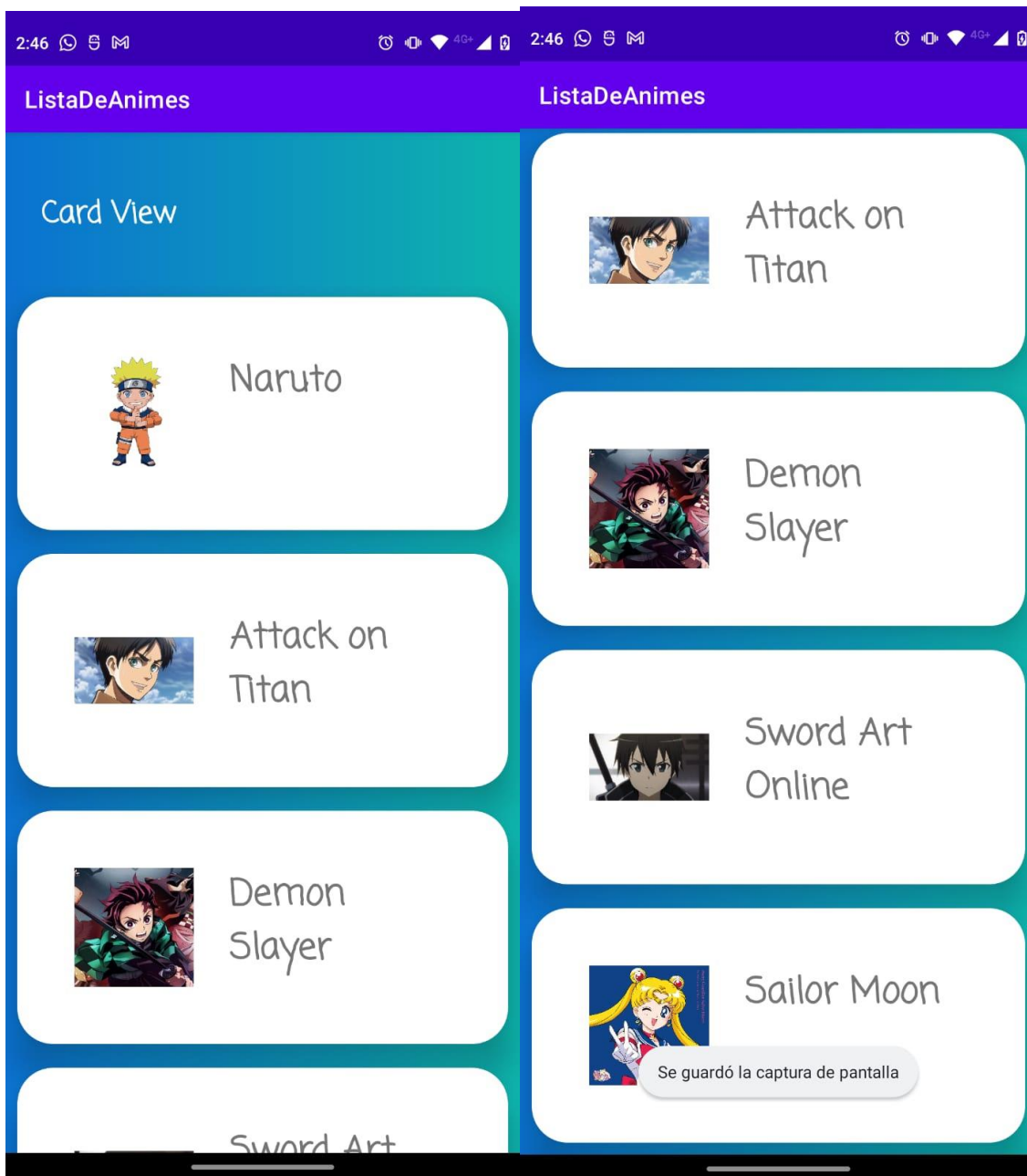
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val recyclerView = findViewById<RecyclerView>(R.id.recyclerView)
        val adapter = AnimeAdapter()

        recyclerView.layoutManager = LinearLayoutManager(this)
        recyclerView.adapter = adapter
    }
}

```

Resultados



Conclusiones

En conclusión, el desarrollo de esta aplicación en Android para mostrar una lista de animes junto con sus imágenes ha sido una experiencia enriquecedora durante mi práctica. He aprendido a utilizar el patrón de adaptador y el componente RecyclerView para mostrar datos de manera eficiente en una lista.

El código presentado demuestra cómo utilizar un adaptador personalizado para vincular datos a las vistas en un RecyclerView. Además, se ha implementado la funcionalidad para mostrar una lista de animes con sus respectivas imágenes en la actividad principal.

En resumen, esta práctica me ha permitido adquirir habilidades importantes en el desarrollo de aplicaciones móviles en Android, específicamente en la implementación de adaptadores personalizados y en el uso de RecyclerViews para mostrar datos de manera eficiente. Considero que esta experiencia ha sido valiosa para mi formación en el desarrollo de aplicaciones móviles y estoy satisfecho con los resultados obtenidos.