

Instituto Tecnológico Superior De Lerdo



Programación móvil

Practica: 4.2

Profesor: Jesús Salas Marín

Alumno: Luis Andres Rodriguez Campos

Carrera: Ing. Informática

Sección: A

Grado: 8

Numero de control: 17231573

Primero se añadira el siguiente código al MainActivity

```
package com.example.pokedex

import ...

class MainActivity : AppCompatActivity(), ListFragment.PokemonSelectListener {
    private lateinit var detailFragment: DetailFragment

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

    override fun onPokemonSelected(pokemon: Pokemon) {
        findNavController(R.id.main_navigation_container).navigate(ListFragmentDirections.actionListFragmentToPokemonDetailFragment(pokemon))
    }
}
```

Ahora se añadira una nueva clase llamada pokemon donde iran todas las características del pokemon

```
package com.example.pokedex

import ...

@Parcelize
data class Pokemon (val id: Long, val name: String, val hp: Int, val attack: Int,
    val defense: Int, val speed: Int, val type: PokemonType, val imageUrl: String, val soundId: Int) : Parcelable {
    enum class PokemonType {
        GRASS, FIRE, WATER, FIGHTER, ELECTRIC
    }
}
```

También crearemos una clase pokemonadapter para mostrar todos los datos del pokemon

```
class PokemonAdapter: ListAdapter<Pokemon, PokemonAdapter.ViewHolder>(DiffCallBack) {
    companion object DiffCallBack : DiffUtil.ItemCallback<Pokemon>() {
        override fun areItemsTheSame(oldItem: Pokemon, newItem: Pokemon): Boolean {
            return oldItem.id == newItem.id
        }

        override fun areContentsTheSame(oldItem: Pokemon, newItem: Pokemon): Boolean {
            return oldItem == newItem
        }
    }
}
```

Como podemos observar se utilizan las características del pokemon

```
lateinit var onItemClickListener: (Pokemon) -> Unit

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PokemonAdapter.ViewHolder {
    val view = LayoutInflater.from(parent.context).inflate(R.layout.pokemon_list_item, parent, attachToRoot: false)
    return ViewHolder(view)
}

override fun onBindViewHolder(holder: PokemonAdapter.ViewHolder, position: Int) {
    val pokemon = getItem(position)
    holder.bind(pokemon)
}

inner class ViewHolder(private val view: View) : RecyclerView.ViewHolder(view) {
    private val idText = view.findViewById<TextView>(R.id.pokemon_id)
    private val nameText = view.findViewById<TextView>(R.id.pokemon_name)
    private val typeImage = view.findViewById<ImageView>(R.id.pokemon_type_image)
```

```
    val imageId = when (pokemon.type) {
        Pokemon.PokemonType.GRASS -> R.drawable.plant
        Pokemon.PokemonType.FIRE -> R.drawable.fire
        Pokemon.PokemonType.WATER -> R.drawable.water
        Pokemon.PokemonType.FIGHTER -> R.drawable.fire
        Pokemon.PokemonType.ELECTRIC -> R.drawable.electric
    }
```

```
    typeImage.setImageResource(imageId)
```

```
    view.setOnClickListener { it: View?
        if (::onItemClickListener.isInitialized) {
            onItemClickListener(pokemon)
        }
    }
```

Se mostraran de manera más detallada los diferentes características de los pokemones

```
class pokemonDetailFragment : Fragment() {

    private val args: pokemonDetailFragmentArgs by navArgs()

    private lateinit var imageView: ImageView
    private lateinit var hpText: TextView
    private lateinit var attackText: TextView
    private lateinit var defenseText: TextView
    private lateinit var speedText: TextView
    private lateinit var loadingWheel: ProgressBar

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val rootView = inflater.inflate(R.layout.fragment_pokemon_detail, container, attachToRoot: false)

        val pokemon = args.pokemon
        imageView = rootView.findViewById(R.id.fragment_detail_image)
        hpText = rootView.findViewById(R.id.fragment_detail_hp)
        attackText = rootView.findViewById(R.id.fragment_detail_attack)
        defenseText = rootView.findViewById(R.id.fragment_detail_defense)
        speedText = rootView.findViewById(R.id.fragment_detail_speed)
        loadingWheel = rootView.findViewById(R.id.loading_wheel)

        val nameText = rootView.findViewById<TextView>(R.id.fragment_detail_name)
        nameText.text = pokemon.name
    }
}
```

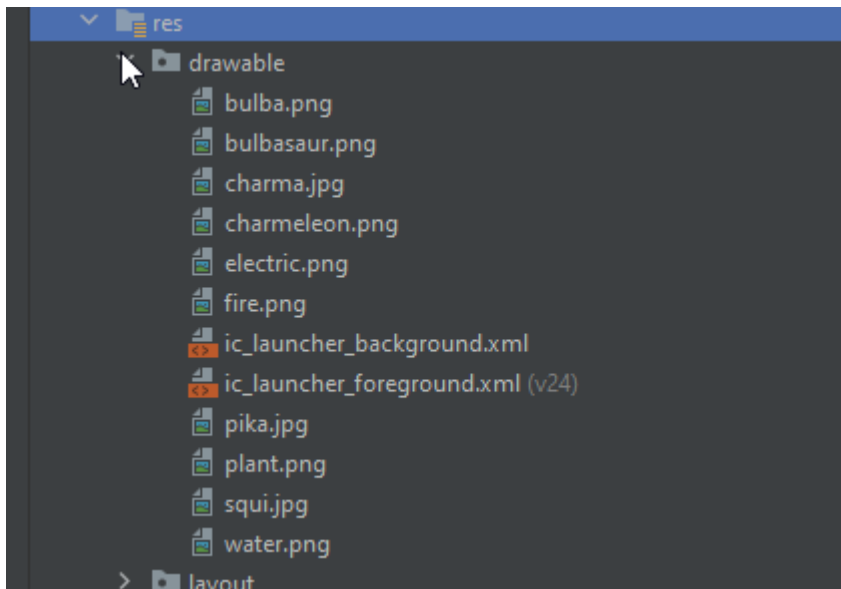
```
private fun setPokemonData(pokemon: Pokemon){
    loadingWheel.visibility= View.VISIBLE
    Glide.with( fragment: this).load(pokemon.imageUrl).listener(object: RequestListener<Drawable> {
        override fun onLoadFailed(
            e: GlideException?,
            model: Any?,
            target: Target<Drawable>?,
            isFirstResource: Boolean
        ): Boolean {
            loadingWheel.visibility= View.GONE
            imageView.setImageResource(R.drawable.squi)
            return false
        }

        override fun onResourceReady(
            resource: Drawable?,
            model: Any?,
            target: Target<Drawable>?,
            dataSource: DataSource?,
            isFirstResource: Boolean
        ): Boolean {
            loadingWheel.visibility= View.GONE
            return false
        }
    }).error(R.drawable.squi).into(imageView);

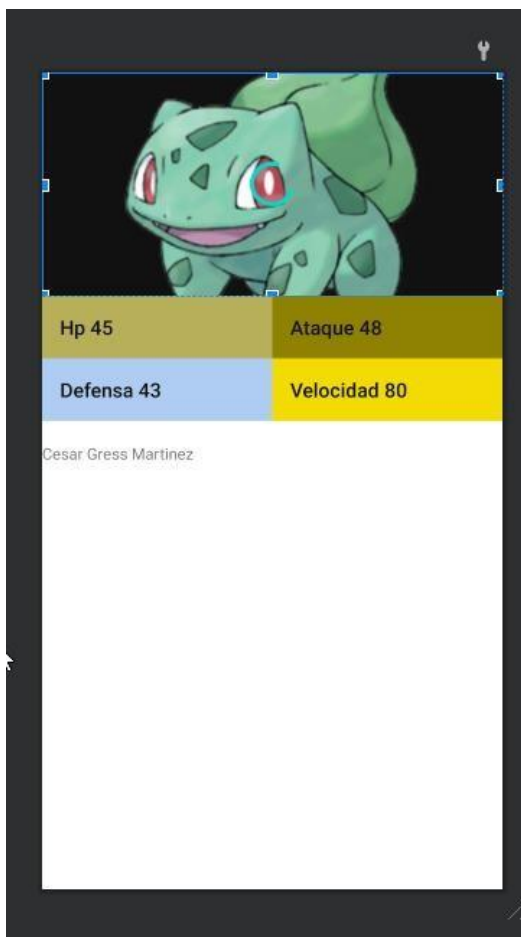
    hpText.text = "Hp: {pokemon.hp}"
    attackText.text = "Ataque: {pokemon.attack}"
    defenseText.text = "Defensa: {pokemon.defense}"
    speedText.text = "Velocidad: {pokemon.speed}"

    val mediaPlayer = MediaPlayer.create(requireActivity(), pokemon.soundid)
    mediaPlayer.start()
}
```

Añadiremos las imágenes necesarias



Y nos movemos hacia lo visual



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".DetailFragment">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/fragment_detail_image"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop"
            android:adjustViewBounds="true"
            android:src="@drawable/bulba"/>

        <ProgressBar
            android:id="@+id/loading_wheel"
            android:layout_gravity="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

    </FrameLayout>

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

```

```

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

    <TextView
        android:id="@+id/fragment_detail_hp"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:padding="16dp"
        android:background="@color/hp_background"
        android:textColor="@color/text_black"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        android:text="Hp 45"/>

    <TextView
        android:id="@+id/fragment_detail_attack"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:padding="16dp"
        android:background="@color/attack_background"
        android:textColor="@color/text_black"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        android:text="Ataque 48"/>

</LinearLayout>

```

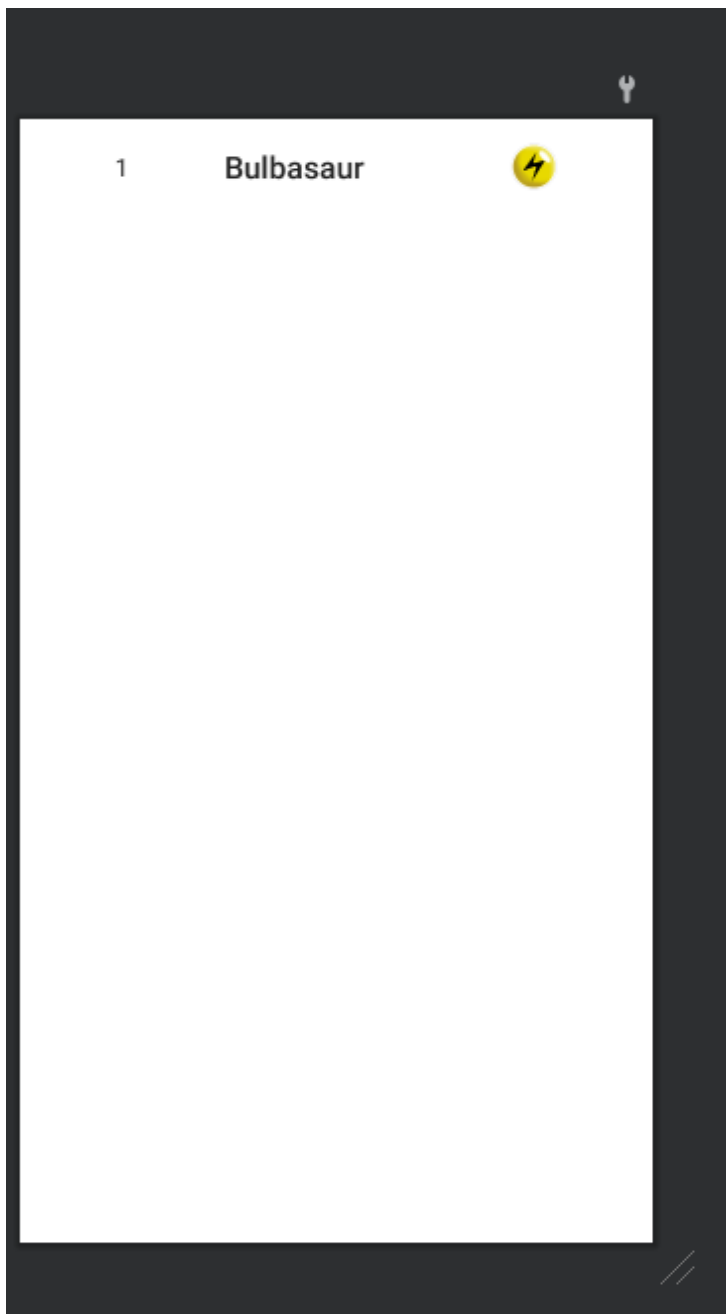
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/fragment_detail_defense"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:padding="16dp"
        android:background="@color/defense_background"
        android:textColor="@color/text_black"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        android:text="Defensa 43"/>

    <TextView
        android:id="@+id/fragment_detail_speed"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:padding="16dp"
        android:background="@color/speed_background"
        android:textColor="@color/text_black"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        android:text="Velocidad 80"/>

</LinearLayout>
```

Ahora un view para mostrar todos los pokemones de la lista




```
?xml version="1.0" encoding="utf-8"?>
```

1 1 ^ v

LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="wrap_content"
xmlns:tools="http://schemas.android.com/tools"
android:gravity="center_vertical"
android:paddingTop="16dp"
android:paddingBottom="16dp"
android:orientation="horizontal">
```

<TextView

```
    android:id="@+id/pokemon_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.S
    tools:text="1"/>
```

<TextView

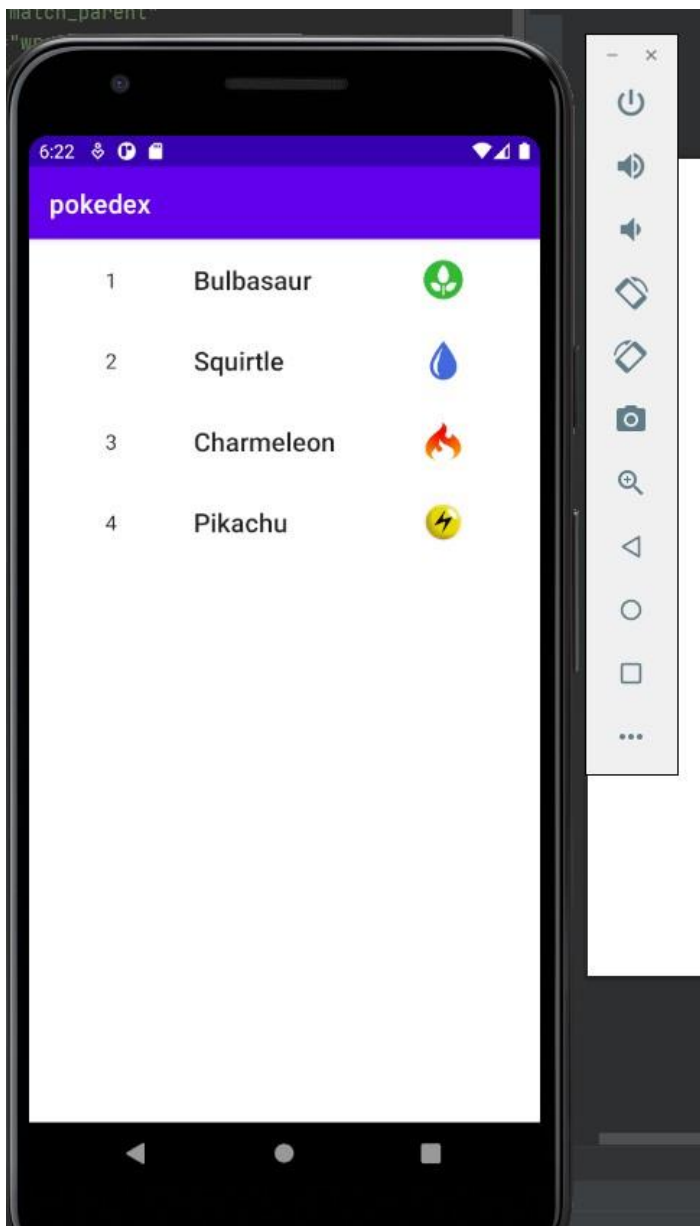
```
    android:id="@+id/pokemon_name"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.T
    tools:text="Bulbasaur"/>
```

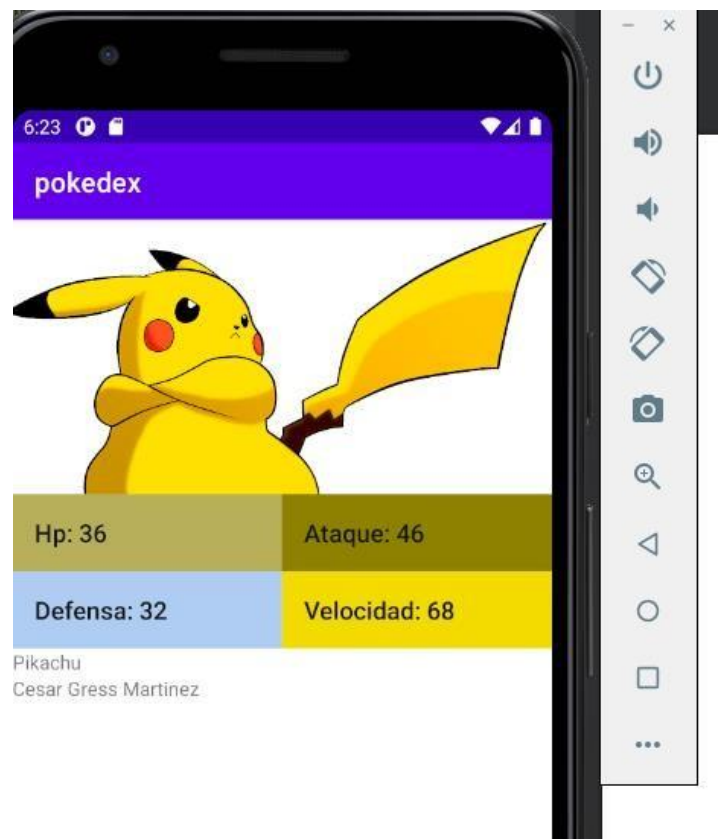
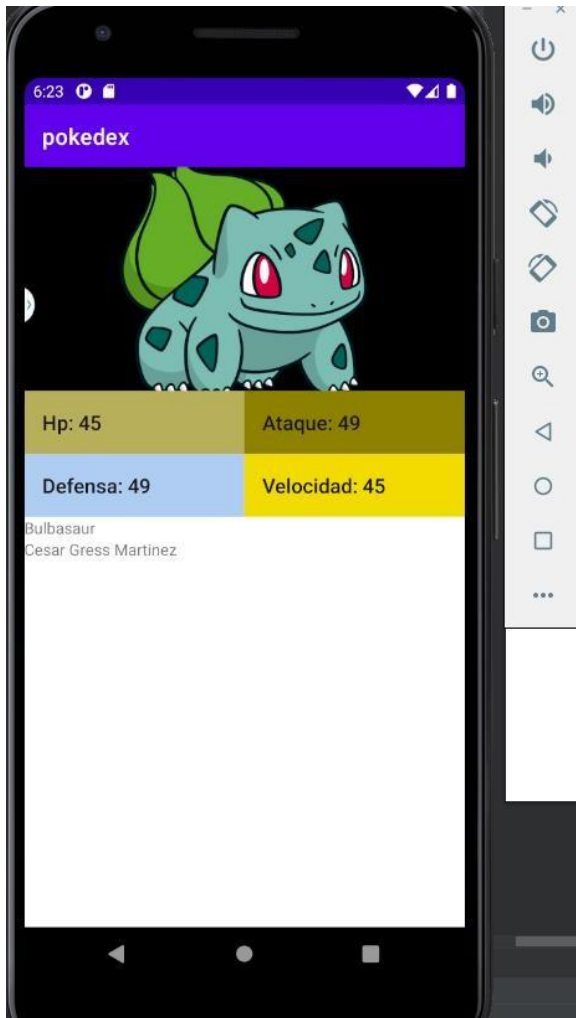
<ImageView

```
    android:id="@+id/pokemon_type_image"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_weight="1"
    tools:src="@drawable/electric"/>
```

/LinearLayout>

Así se ve la aplicación en funcionamiento





Conclusiones

Para la aplicación se usaron conocimientos que se aprendieron con anterioridad en practicas pasadas, aunque ahora se utilizaron extras como sonidos de los pokemons y los fragment, esto es para la utilización de layout con más facilidad y el uso de datos en diferentes layouts, también se aprendió el manejo de imágenes por url, ya que es más fácil usar imágenes de internet que agregándolas manualmente, ya que solo se usan iconos en algunas views.