



Instituto Politécnico Nacional  
Escuela Superior de Cómputo



Desarrollo de Aplicaciones Móviles Nativas

## **“Lista con imágenes (frutas)”**

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 artículos con sus detalles, imágenes y precios. En este informe, presentaré el código relevante y explicaré cómo se implementaron las funcionalidades clave.

El código principal se divide en tres clases: `AdaptadorArticulo`, `Articulo` y `MainActivity`. La clase `Articulo` representa un artículo con su nombre, precio, descripción e imagen. La clase `VistaController` se utiliza para gestionar las vistas de un elemento de la lista de artículos. Por último, en la actividad principal (`MainActivity`), se configura el `ListView` y se asigna un adaptador personalizado (`AdaptadorArticulo`) para mostrar la lista de artículos.

En el adaptador `AdaptadorArticulo`, se implementan los métodos necesarios para gestionar la visualización de los elementos de la lista en el `ListView`. Se infla la vista para cada elemento y se asignan los valores adecuados a las vistas correspondientes, como el nombre del artículo, la descripción, la imagen y el precio. Además, se define un listener de clics en los elementos de la lista para mostrar un mensaje con el artículo seleccionado.

En la actividad principal (`MainActivity`), se configura el `ListView` y se asigna un adaptador personalizado (`AdaptadorArticulo`) que se instancia con una lista ficticia de artículos generados en el método `generarDatosFicticios()`. Estos artículos incluyen nombres, precios, descripciones e imágenes de ejemplo. Al finalizar la configuración, se notifica al adaptador que los datos han cambiado mediante `adaptadorDatosArticulo?.notifyDataSetChanged()`.

## Desarrollo

```
class AdaptadorArticulo(private var activity: Activity,
private var items: ArrayList<Articulo>): BaseAdapter(){
    override fun getCount(): Int {
        return items.size
    }

    override fun getItem(position: Int): Any {
        return items[position]
    }

    override fun getItemId(position: Int): Long {
        return position.toLong()
    }

    override fun getView(position: Int, convertView: View?, parent:
ViewGroup?): View {
        val vistaACargos:View?
        val vista: VistaController
        if(convertView == null){
            val inflater =
activity?.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as
LayoutInflater
            vistaACargos =
inflater.inflate(R.layout.elemento_articulo,null)
            vista = VistaController(vistaACargos)
            vistaACargos?.tag = vista
        }else{
            vistaACargos = convertView
            vista = vistaACargos.tag as VistaController
        }
        var articulos = items[position]
        vista.imgArticulo?.setImageResource(articulos.imagenArticulo)
        vista.txtNombre?.text=articulos.nombreArticulo
        vista.txtDetalle?.text=articulos.descripcionArticulo
        vista.txtPrecio?.text= articulos.precioArticulo.toString()

        vistaACargos?.setOnClickListener(){
            Toast.makeText(activity,"Articulo Seleccionado:
${articulos.nombreArticulo}",
            Toast.LENGTH_SHORT).show()
        }
        return vistaACargos as View
    }
}
```

```
class Articulo(val nombreArticulo:String,
                val precioArticulo: Double,
                val descripcionArticulo:String,
                val imagenArticulo:Int) : java.io.Serializable
```

```

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

        var lista=findViewById<ListView>(R.id.lista)
        var adaptadorDatosArticulo = AdaptadorArticulo(this,
generarDatosFicticios())
        lista?.adapter = adaptadorDatosArticulo
        adaptadorDatosArticulo?.notifyDataSetChanged()
    } // fin onCreate

    private fun generarDatosFicticios(): ArrayList<Articulo> {
        var listaDeArticulos = ArrayList<Articulo>()
        val a1:Articulo = Articulo(
            "Fresa",500.0,"Fresa",R.drawable.fresa
        )
        val a2:Articulo= Articulo(
            "Mango",29.3,"Mango",R.drawable.mango
        )
        val a3:Articulo= Articulo(
            "Kiwi",60.5,"Kiwi",R.drawable.kiwi
        )
        val a4:Articulo= Articulo(
            "Sandia",120.34,"Sandia",R.drawable.sandia
        )
        val a5:Articulo= Articulo(
            "Naranja",10.5,"Naranja",R.drawable.naranja
        )

        listaDeArticulos.add(a1)
        listaDeArticulos.add(a2)
        listaDeArticulos.add(a3)
        listaDeArticulos.add(a4)
        listaDeArticulos.add(a5)

        return listaDeArticulos
    }
}












```

```

class VistaController(renglon: View?) {
    var txtNombre:TextView?=null
    var txtDetalle:TextView?=null
    var imgArticulo:ImageView?=null
    var txtPrecio:TextView?=null
    init {
        this.txtNombre = renglon?.findViewById(R.id.nombre)
        this.txtDetalle = renglon?.findViewById(R.id.txtDescripcion)
        this.imgArticulo = renglon?.findViewById(R.id.imageView)
        this.txtPrecio = renglon?.findViewById(R.id.precio)
    }
}

```

# Resultados

ListaArticulos		ListaArticulos	
	Fresa Fresa 500.0		Fresa Fresa 500.0
	Mango Mango 29.3		Mango Mango 29.3
	Kiwi Kiwi 60.5		Kiwi Kiwi 60.5
	Sandia Sandia 120.34		Sandia Sandia 120.34
	Naranja Naranja 10.5		Naranja Naranja 10.5
 Artículo Seleccionado: Mango			

## Conclusiones

En conclusión, el desarrollo de esta aplicación en Android para mostrar una lista de artículos con sus detalles, imágenes y precios ha sido una experiencia valiosa durante mi práctica. He aprendido a utilizar un adaptador personalizado (`AdaptadorArticulo`) para gestionar la visualización de los elementos en un `ListView`.

El código presentado demuestra cómo implementar un adaptador personalizado para asignar valores a las vistas correspondientes en un `ListView`. Además, se ha implementado la funcionalidad para mostrar una lista de artículos con sus detalles, imágenes y precios en la actividad principal (`MainActivity`).

Durante la práctica, también se ha utilizado la clase `Articulo` para representar un artículo con su información básica. Además, se ha creado la clase `VistaController` para facilitar la gestión de las vistas de un elemento de la lista.

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