

# UNIVERSIDAD DON BOSCO

Facultad de Ingeniería  
Escuela de Computación  
Desarrollo de  
Aplicaciones Móviles



## Taller 2

Grupo Teórico: G01T

Apellidos	Nombres	Carné
<u>William Ernesto Ramos Valladares</u>		<u>RV200068</u>
<u>Kenya Elizabeth Parada Palma</u>		<u>pp220664</u>

Para el manejo de la aplicación se uso botones enlazados entre activitys y para el manejo de la base de datos se uso la clase proporcionada por SQL solo se usaron métodos put y get, al momento de buscar un producto por su id elste se muestra en un text view donde se cambia su valor por el nombre y el precio, luego al darle comprar los datos se mandan a la base de datos para luego ser recuperados en otro activity y así manejar las peticiones en lugares distintas, hay cálculos donde se genera el total, la pantalla splash se modifiko para ser lo primero que se vea por una duración definida de tiempo y luego pasaría al siguiente activty que sería el main donde ya se pueden manejar los eventos con mayor accesibilidad

El controlador de la base de datos es este:

```

1  package com.example.farmacia
2
3  import android.content.ContentValues
4  import android.content.Context
5  import android.database.sqlite.SQLiteOpenHelper
6  import android.database.sqlite.SQLiteDatabase
7  import android.database.sqlite.SQLiteDatabase.CursorFactory
8  import androidx.core.content.contentValuesOf
9
10 class sqlHelper(context: Context, name: String, factory: CursorFactory?, version: Int) : SQLiteOpenHelper
11     (context, "tienda.db", null, 1) {
12
13     override fun onCreate(db: SQLiteDatabase) {
14
15         val ordenCreacion = "CREATE TABLE tienda (_id INTEGER PRIMARY KEY AUTOINCREMENT, producto TEXT, cantidad TEXT, total TEXT)"
16         db.execSQL(ordenCreacion)
17     }
18
19     override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
20
21     }
22
23     fun add(producto: String, cantidad: String, total: Double){
24         val datos = ContentValues()
25         datos.put("producto", producto)
26         datos.put("cantidad", cantidad)
27         datos.put("total", total)
28
29         val db = this.writableDatabase
30         db.insert("tienda", null, datos)
31         db.close()
32     }
33 }
34 }

```

```

57
58
59     }
60
61     btnComprar.setOnClickListener{
62         var id = txtid.text.toString().toInt()
63
64         id -= 1
65
66         val total = txtcantidad.text.toString().toInt() * precio[id]
67         if (tvproducto.text.isNotBlank() && txtcantidad.text.isNotBlank()){
68             tiendaDBHelper.add(tvproducto.text.toString(),
69                 txtcantidad.text.toString(),
70                 total)
71             txtid.text.clear()
72             tvproducto.text="Producto"
73             txtcantidad.text.clear()
74             Toast.makeText(this, "Compra exitosa, total: $ $total", Toast.LENGTH_SHORT).show()
75
76         }
77         else{
78             Toast.makeText(this,"No se pudo comprar", Toast.LENGTH_LONG).show()
79         }
80     }

```

Y el controlador de eventos definido en un botón es este qe si las casillas están llenas manda un mensaje y hace la petición put si no da error y no deja continuar