

1. El menú de opciones (tres puntitos de la parte superior derecha de la pantalla) muestra una única opción “Añadir a la lista”.

a. Al pulsar en “Añadir a la lista” se muestra un cuadro de dialogo personalizado que pide al usuario los siguientes datos:

- Producto
- Cantidad
- Precio Aproximado por unidad

```
private void abrirDialogo(Context context){  
    LayoutInflater li = LayoutInflater.from(context);  
    View layoutCuadro = li.inflate(R.layout.dialogo, root: null);  
  
    AlertDialog.Builder cuadroDialogo = new AlertDialog.Builder(context);  
    cuadroDialogo.setView(layoutCuadro);  
  
    final EditText edit_producto = (EditText) layoutCuadro.findViewById(R.id.edit_producto);  
    final EditText edit_cantidad = (EditText) layoutCuadro.findViewById(R.id.edit_cantidad);  
    final EditText edit_precio = (EditText) layoutCuadro.findViewById(R.id.edit_precio);  
  
    cuadroDialogo  
        .setCancelable(false)  
        .setPositiveButton( text: "Añadir",  
            new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog,int id) {  
                    Producto producto = new Producto(edit_producto.getText().toString(),  
                        Integer.valueOf(edit_cantidad.getText().toString()),  
                        Float.parseFloat(edit_precio.getText().toString()));  
                    long id_producto = bdAccesoDatos.insertar(producto);  
                    Toast toast;  
                    if (id_producto != -1) {  
                        toast = Toast.makeText(context,  
                            text: "Producto creado con éxito, id: "+id_producto, duration: 2000);  
                    } else {  
                        toast = Toast.makeText(context,  
                            text: "Ha ocurrido un error, el producto no se pudo crear", duration: 2000);  
                    }  
                    toast.show();  
                    bdAccesoDatos.consultar(productos);  
                    ordenarLista();  
                }  
            })  
        .setNegativeButton( text: "Cancelar",  
            new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog,int id) {  
                    dialog.cancel();  
                }  
            })  
        .show();  
  
    AlertDialog alertDialog = cuadroDialogo.create();  
    cuadroDialogo.show();  
}
```

Al pulsar el botón “Añadir a la lista”, se ejecuta esta función. Crea un diálogo personalizado, con el layout que hemos creado previamente con el formulario. Se recogerán los dos eventos para los botones. En el caso de pulsar “Añadir” se creará un objeto de tipo Producto y insertará en la base de datos. Llamaremos la función ordenarLista, para ordenar y refrescar el adapter.

```

public void ordenarLista() {

    Collections.sort(productos, new Comparator<Producto>() {

        @Override
        public int compare(Producto producto1, Producto producto2) {
            return producto1.getProducto().compareToIgnoreCase(producto2.getProducto());
        }

    });
    adaptador.notifyDataSetChanged();
}

```

b. El cuadro de diálogo anterior tiene dos botones:

- Añadir. Guarda los datos introducidos por el usuario en una Base de Datos SQLite.
 - El producto será un String, la cantidad será un entero y el precio será un float.
 - La Base de Datos debe de tener un campo distinto a los anteriores que actúe como clave primaria.
 - Al pulsar el botón “Añadir”, se muestra un mensaje fiable en el que se mostrará el id del registro que se acaba de añadir.

```

public class Producto {

    private Integer id;
    private String producto;
    private Integer cantidad;
    private Float precio;

    public Producto(Integer id, String producto, Integer cantidad, Float precio) {
        this.id = id;
        this.producto = producto;
        this.cantidad = cantidad;
        this.precio = precio;
    }

    public Producto(String producto, Integer cantidad, Float precio) {
        this.producto = producto;
        this.cantidad = cantidad;
        this.precio = precio;
    }

}

```

Estos son los tipos de los atributos del objeto producto.

- Cancelar. Cierra el cuadro de diálogo sin guardar los datos.

2. ListView:

a. El ListView de la Activity principal muestra al inicio de la aplicación todos los elementos que hay en la lista de la compra.

```

lista = findViewById(R.id.listaCompra);

bdAccesoDatos = new BDAccesoDatos(c: this);

bdAccesoDatos.consultar(productos);
adaptador = new ArrayAdapter<>(getApplicationContext(),android.R.layout.simple_list_item_1,productos);
lista.setAdapter(adaptador);
ordenarLista();

```

Se consultan los productos existentes en la base de datos para crear el adaptador del listView.

b. Cuando se inserta un nuevo elemento, la lista se actualiza de forma automática mostrando la nueva información.

Esto se hace utilizando notifyDataSetChanged().

c. Al hacer una pulsación corta sobre un elemento de la lista, se abre una nueva Activity con todos los datos del mismo (id, producto, cantidad, y precio).

```

lista.setClickable(true);
lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int position, long arg3) {
        Intent intent = new Intent( packageContext: MainActivity.this, ResumeActivity.class);
        intent.putExtra( name: "id", productos.get(position).getId());
        intent.putExtra( name: "producto", productos.get(position).getProducto());
        intent.putExtra( name: "cantidad", productos.get(position).getCantidad());
        intent.putExtra( name: "precio", productos.get(position).getPrecio());
        startActivity(intent);
    }
});

```

Se obtiene el producto, pulsado a partir de la posición y consultándolo en el arrayList. Se envía la información en el intent a la nueva activity.

- La nueva activity no puede hacer ninguna consulta a la Base de Datos, es decir, la activity principal enviará toda la información que se necesite mostrar en la activity secundaria.
- Esta activity secundaria tiene un Botón “Atrás” que lleva de nuevo a la activity principal.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_resume);

    Intent intent = getIntent();
    Integer id = intent.getIntExtra( name: "id", defaultValue: 0);
    String producto = intent.getStringExtra( name: "producto");
    Integer cantidad = intent.getIntExtra( name: "cantidad", defaultValue: 0);
    Float precio = intent.getFloatExtra( name: "precio", (float) 0.0);

    edit_id = findViewById(R.id.edit_id);
    edit_producto = findViewById(R.id.edit_producto);
    edit_cantidad = findViewById(R.id.edit_cantidad);
    edit_precio = findViewById(R.id.edit_precio);
    btn_atras = findViewById(R.id.atras);

    edit_id.setText(id.toString());
    edit_producto.setText(producto);
    edit_cantidad.setText(cantidad.toString());
    edit_precio.setText(precio.toString());

    btn_atras.setOnClickListener( new View.OnClickListener() {
        @Override
        public void onClick(View v) { finish(); }
    });
}

```

En la nueva actividad se obtiene la información contenida en el intent y se muestra. Se añade el evento de click para el botón y finaliza la activity para volver atrás.

d. Al hacer una pulsación larga sobre un elemento del ListView, se debe borrar el elemento, tanto de la Base de Datos como de la lista.

- Mostrar un cuadro de diálogo que pida la confirmación de la eliminación.
- Mostrar un mensaje fiable de que si el borrado se ha realizado correctamente.

```

lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public boolean onItemClick(AdapterView<?> arg0, View arg1,
                                int pos, long id) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context.MainActivity.this);
        builder.setMessage("¿Seguro que desea borrar el producto?")
        .setPositiveButton( text: "Si", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                long cant = bdAccesoDatos.eliminar(productos.get(pos).getId());
                Toast toast;
                if (cant == 1) {
                    toast = Toast.makeText(getApplicationContext(), text: "El producto fue eliminado con éxito", duration: 2000);
                } else {
                    toast = Toast.makeText(getApplicationContext(), text: "Ha ocurrido un error, el producto no se pudo eliminar", duration: 2000);
                }
                toast.show();
                productos.remove(pos);
                ordenarLista();
            }
        })
        .setNegativeButton( text: "No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        // Create the AlertDialog object and return it
        builder.create();
        builder.show();
        return true;
    }
});

```

Se añade el evento de pulsación larga para el listView, se muestra un dialogo de confirmación y se elimina de base de datos el producto y del ArrayList. Posteriormente se actualiza el adapter.