

# Programación en el Entorno Android

Master en Ingeniería Informática - Programación de Dispositivos Móviles

## Práctica 3

### Introducción

Deberá desarrollar una aplicación que gestione una base de clientes, productos y pedidos, siguiendo el modelo de datos y la interfaz que se muestran en los siguientes apartados.

### Modelo de datos

El modelo de datos de la aplicación, que se muestra en la Ilustración 1, consta de tres entidades: *Customer*, *Product* y *order*. La tabla *Customer* tendrá además un *IDCustomer* (Primary Key), la tabla *Product* tendrá además un *IDProduct* (Primary Key) y la tabla *Order* tendrá además un *IDOrder* (Primary Key) un *IDCustomer* (Foreign Key) y un *IDProduct* (Foreign Key).

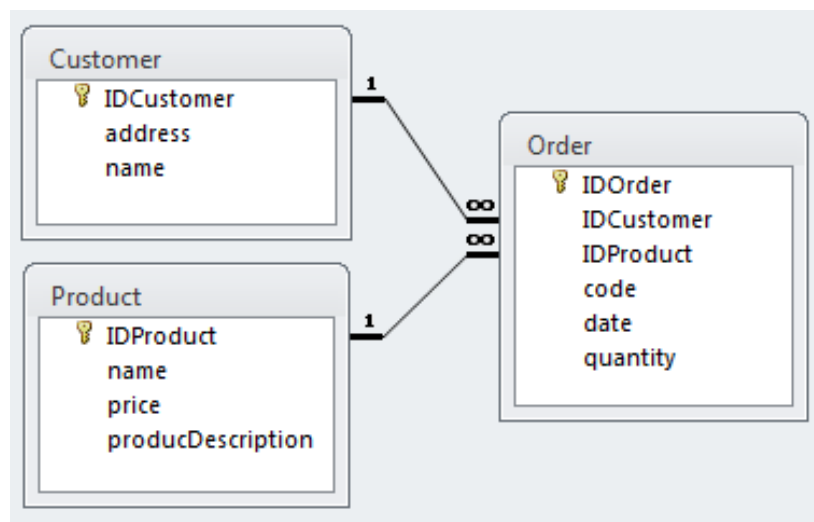


Ilustración 1

La entidad *Customer* representa un cliente y tiene tres atributos:

- Atributo *idcustomer*: de tipo *Integer*, representa la primarykey
- Atributo *address*: de tipo *String*, representa la dirección del cliente.
- Atributo *name*: de tipo *String*, representa el nombre del cliente.

La entidad *Product* representa un producto y tiene cuatro atributos:

- Atributo *idproduct*: de tipo *Integer*, representa la primarykey
- Atributo *name*: de tipo *String*, representa el nombre del producto.
- Atributo *price*: de tipo *Decimal*, representa el precio del producto.
- Atributo *productDescription*: de tipo *String*, representa una descripción breve del producto.

La entidad *Order* representa un pedido de una cierta cantidad de un producto efectuado por un cliente y tiene los siguientes atributos:

- Atributo *idorder*: de tipo *Integer*, representa la primarykey
- Atributo *idcustomer*: de tipo *Integer*, representa una foreignkey
- Atributo *idproduct*: de tipo *Integer*, representa la foreignkey
- Atributo *code*: de tipo *String*, representa el código del pedido.
- Atributo *date*: de tipo *Date*, representa la fecha de realización del pedido.
- Atributo *quantity*: de tipo *Integer 16*, representa el número de unidades del producto solicitadas por el pedido.

Además, el modelo establece las siguientes relaciones:

Una relación de uno a muchos entre las entidades *Customer* y *Order*, que representa el conjunto de pedidos realizados por un cliente, y su inversa, el cliente que ha realizado un pedido.

Una relación uno a muchos entre las entidades *Product* y *Order*, que representa el conjunto de pedidos para un producto, y su inversa, el producto que se solicita en un pedido.

Se debe usar la librería ROOM DATABASE para la realización de la práctica.

## Interfaz

La interfaz de la aplicación tiene tres pestañas que conducen, cada una, a una sección distinta. Se debe implementar una **navegación SwipeViews** con las tres pestañas (*Customers*, *Products* y *Orders*) seleccionables. Se debe implementar un **Menú Lateral Izquierdo** que contenga los mismos nombres de las pestañas que llevarán a cabo las mismas acciones. Para el diseño usar los iconos de google de <https://design.google.com/icons/> siempre que se pueda. Para todos los textos, dimensiones, valores, etc. usar los recursos adecuadamente.

## PestañaCustomers

Esta sección está formada por dos vistas que gestionan los clientes (entidad *Customer*). La primera vista de esta secuencia (Ilustración 2) es un *RecyclerView* que muestra el listado de clientes. El *RecyclerView* presenta una configuración lineal en donde cada fila muestra información sobre un cliente: nombre y dirección. Las filas están ordenadas por el nombre del cliente y presentan el símbolo "chevronright" a la derecha para indicar que se puede acceder, mediante un click, a la información del cliente para modificarla en una segunda vista.

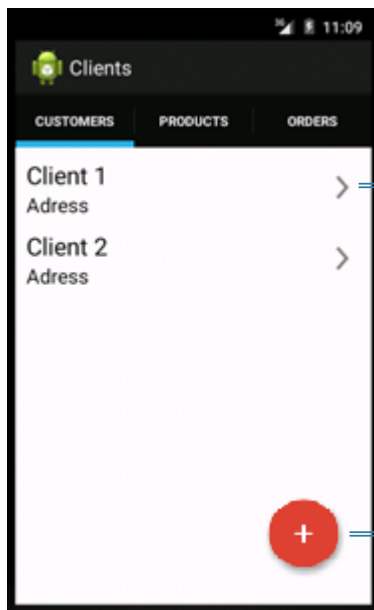


Ilustración 2

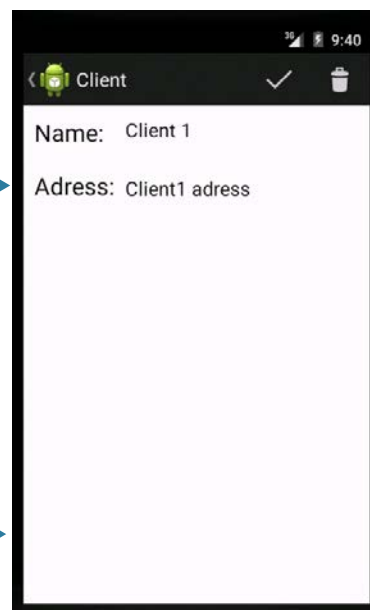


Ilustración 3

En la *Barra App* de la primera vista aparece el título. Se debe implementar el botón flotante para añadir más clientes a la base de datos. Al presionar este botón se accede a la segunda vista vacía que permite introducir los datos para un nuevo cliente. Si se realiza una presión larga sobre un cliente se preguntará en un diálogo emergente si se desea eliminar.

En la Barra app de la segunda pantalla hay dos opciones:

- Si se llega desde el signo más de la barra app o desde el botón flotante. Debe aparecer, además del título, un botón a la derecha "check" que permite grabar el nuevo cliente en la base de datos. No debe aparecer la papelera. La Barra App se debe configurar para que permita volver a la pantalla anterior. Si se desea salir de esta vista sin guardar el nuevo cliente, se debe tocar el botón up de la Barra App y volverá a la vista anterior o bien el botón retroceder el móvil.
- Si se llega seleccionando un elemento de la lista. Debe aparecer, además del título, dos botones, el de grabar "check" y la papelera "delete", que permitirá borrar el elemento. Las acciones para cancelar la edición y volver atrás son las mismas que en el apartado anterior.

El botón de borrar deberá confirmar la acción con un diálogo emergente y mostrar un aviso emergente (Toast) informando que la acción se llevó a cabo correctamente o no. A la hora de eliminar se debe mantener la coherencia de la base de datos, es decir, no se puede eliminar un cliente sin tiene órdenes de compra asociadas.

El botón de grabar deberá mostrar un aviso emergente (Snackbar) con un mensaje indicando que la acción se ha llevado cabo y una acción de deshacer.

La segunda vista debe permitir retroceder a la pantalla anterior sin llevar a cabo ninguna acción, ni de eliminación ni de grabación, para ello se habilitará el botón UP de la *Barra App*.

### Pestaña Products

Esta pestaña tiene el mismo funcionamiento que la pestaña clientes, pero con los datos de la tabla productos.

### PestañaOrders

Esta sección tiene dos vistas igual que las anteriores. El *RecyclerView* mostrará en cada fila el nombre del cliente, el nombre del producto y el código de la orden, ordenados por cliente, producto y código.

En la segunda vista, la fecha del pedido se muestra en un *Date Picker* que, si se está insertando un nuevo pedido ofrece como opción inicial la fecha actual. Los campos para editar el cliente y el producto no se pueden editar manualmente. El campo cantidad se puede editar usando un teclado numérico, mientras que los otros dos (cliente y producto) se asocian con sendos botones "search" que muestran un nuevo *RecyclerView* (de pantalla completa) en la que se puede seleccionar un cliente o un producto, según corresponda. Al seleccionar un elemento (cliente o producto) automáticamente vuelve a la vista anterior y rellena los campos correspondientes. Si se está editando un pedido existente, esta tabla mostrará seleccionado inicialmente el cliente o el producto actualmente asociados con dicho pedido. El precio de la orden se calcula automáticamente, a partir del precio unitario del producto y la cantidad de unidades consignada en la orden, con dos decimales.

Se debe mantener la coherencia de los datos en todo momento y no permitir almacenar un registro con campos vacíos, ni acceder a pantallas que no tienen sentido por carecer de datos.

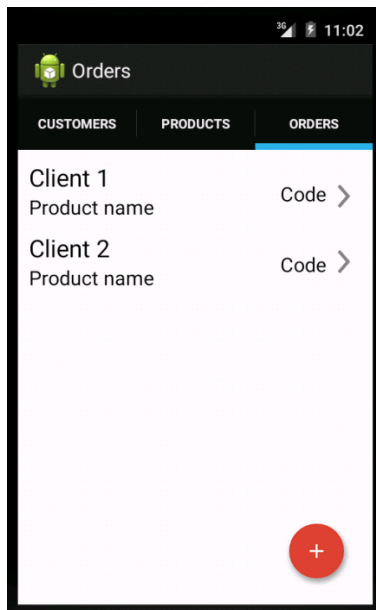


Ilustración 4

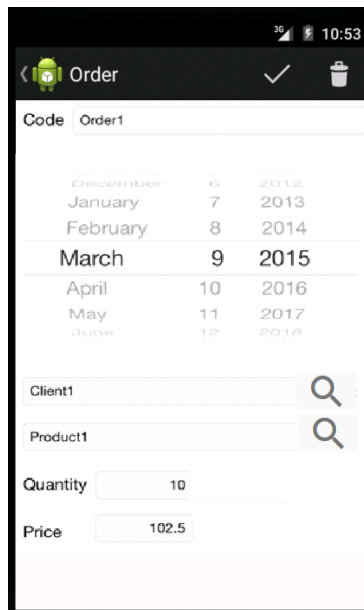


Ilustración 5

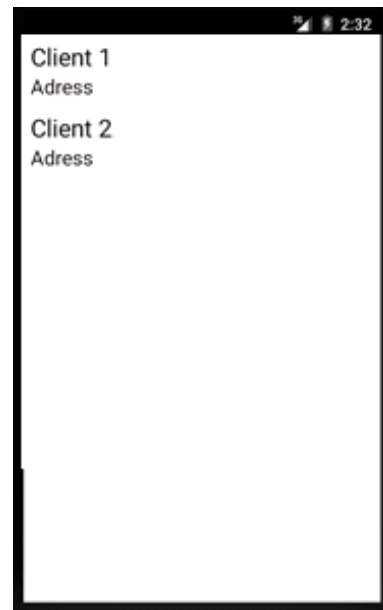


Ilustración 6

## Resultado

Como resultado de la práctica, se entregará un fichero comprimido que contendrá el proyecto de la práctica completo, con todas sus carpetas y archivos, listo para ser usado. Por favor, usar el siguiente formato para el nombre del proyecto y el fichero comprimido NombreDelAlumno\_PrácticaNum