PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

TEMA 6 - ACTIVIDAD 2 - GESTIÓN BANCARIA

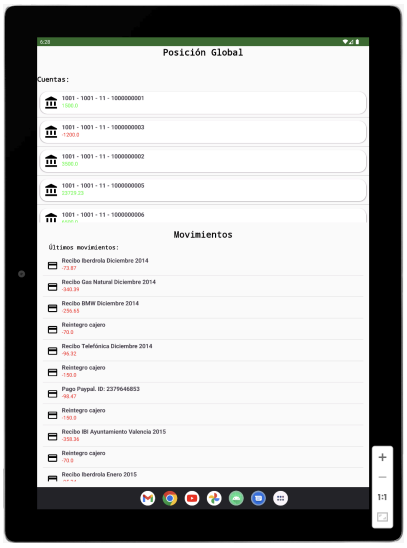
Actividad Fragments

Debéis adaptar la aplicación del banco para que utilice Fragments, de forma que se pueda visualizar para las siguientes configuraciones:

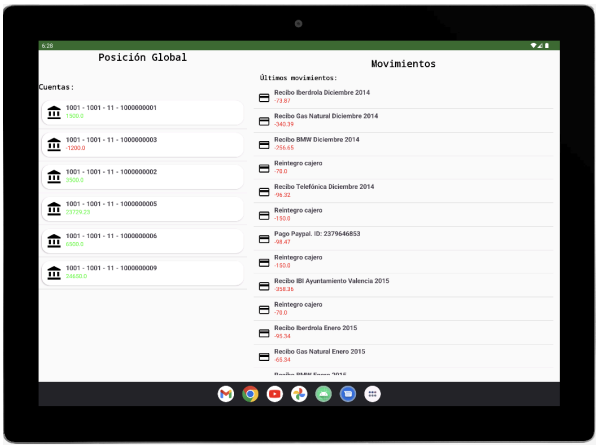
- Móvil
- Tablet (tamaño large) en vertical
- Tablet (tamaño large) en horizontal

Para ello vamos a realizar dicha adaptación a las pantallas de **Posición global** donde un fragment mostrará el listado de las cuentas del usuario y cuando seleccionemos una cuenta nos detalla todos sus movimientos mediante otro fragment.

Este comportamiento será el mismo que la segunda opción del menú Movimientos, donde habíamos utilizado un Spinner para seleccionar la cuenta y se actualizarán sus movimientos. Así tendremos dos formas de realizar lo mismo, una utilizando fragments y otra con el componente Spinner visto en el tema anterior.

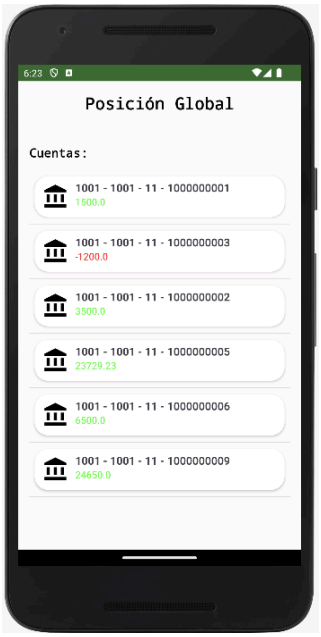


Tablet en vertical

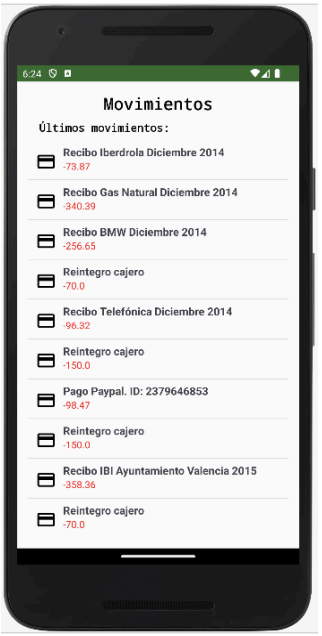


Tablet en horizontal

- En las tablets se utiliza 1 Activity donde se visualizan los 2 fragments.



GlobalPositionActivity



GlobalPositionDetailsActivity

- En los móviles se utilizan 2 Activity donde cada una contiene 1 fragment.

Para intentar realizar todos una solución parecida, vamos a tener en cuenta:

- La configuración de las pantallas será la misma que la aplicación de ejemplo del correo electrónico de los apuntes del tema 6, solo que tendréis que adaptarla al banco.
  - activities: GlobalPositionActivity, GlobalPositionDetailsActivity.
  - adapters: AccountsAdapter, AccountsListener, MovementsAdapter.
  - fragments: AccountsFragment, AccountsMovementsFragment.
  - layout:
    - activity_global_position.xml * 3 (normal, large-land, large-port)
    - activity_global_position_details.xml
    - fragment_accounts.xml, fragment_accounts_movements.xml
    - item_list_accounts.xml, item_list_movements.xml

Para pasar información de una actividad a un fragment, lo haremos de la siguiente forma:

AccountsFragment.kt

```
...
companion object {
   @JvmStatic
   fun newInstance(c: Cliente) =
       AccountsFragment().apply {
           arguments = Bundle().apply {
               putSerializable(ARG_CLIENTE, c)
           }
       }
```

```
}


override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    arguments?.let {
        cliente = it.getSerializable(ARG_CLIENTE) as Cliente
    }
}


    ...
```

GlobalPositionActivity.kt

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityGlobalPositionBinding.inflate(layoutInflater)
    setContentView(binding.root)

    //Recuperar el cliente
    val clienteLogeado = intent.getSerializableExtra("Cliente")

    //Creamos la instancia del fragment
    val frgAccounts: AccountsFragment = AccountsFragment.newInstance(clienteLogeado as
Cliente)

    ...
```

Cuando modificamos la lógica del fragment (AccountsFragment.kt) creamos o modificamos el método newInstance() para poder pasarle un objeto desde otra clase (activity) y también modificamos el método onCreate() para recuperar el objeto u objetos que nos pasan. Ahora solamente nos queda crear un objeto de tipo fragment desde una actividad con el método newInstance() y el objeto que queramos.

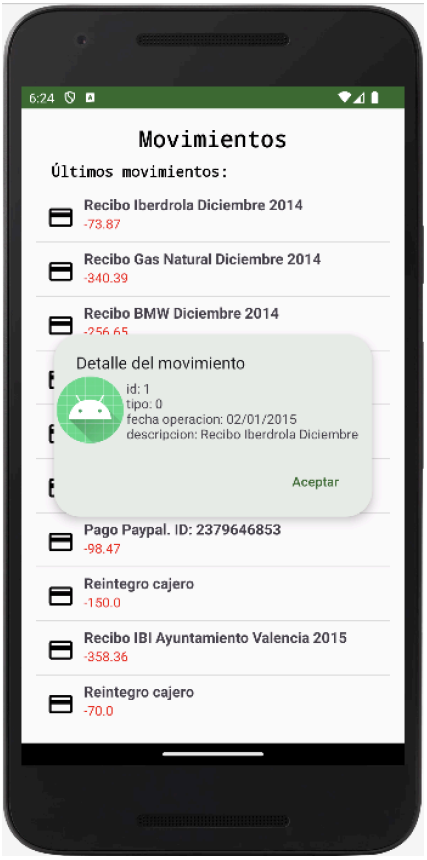
De forma análoga se hace para pasar datos a AccountsMovementsFragment.kt.

Nota: al utilizar este modo de pasar información entre la Activity y el fragment, nos vemos obligados a utilizar los fragments dinánimos, así que asegúrate que no esté el atributo name en el xml.

Actividad Diálogo

En segundo lugar, tenéis que diseñar un diálogo personalizado que siga el tema (colores, fuente, diseño, etc) de vuestra aplicación de banco y que, cuando se pulse en uno de los movimientos, muestre toda la información del movimiento: importe, descripción, fecha, datos de la cuenta de origen, datos de la cuenta destino…

Para esta segunda opción crearemos un layout con el nombre de dialog_movement.xml.



MovementDialog

Se ha de subir un fichero zip a la plataforma, que llevará de nombre

```
t6a2-PrimerApellido-Nombre.zip
```

Y subirlo a la plataforma.