



Universidad de colima
Facultad de Telemática
Ingeniería de software
Programación de Móviles
Sergio García Michel
Proyecto final
“App para apoyo de comercios locales”

5E

Colima col.

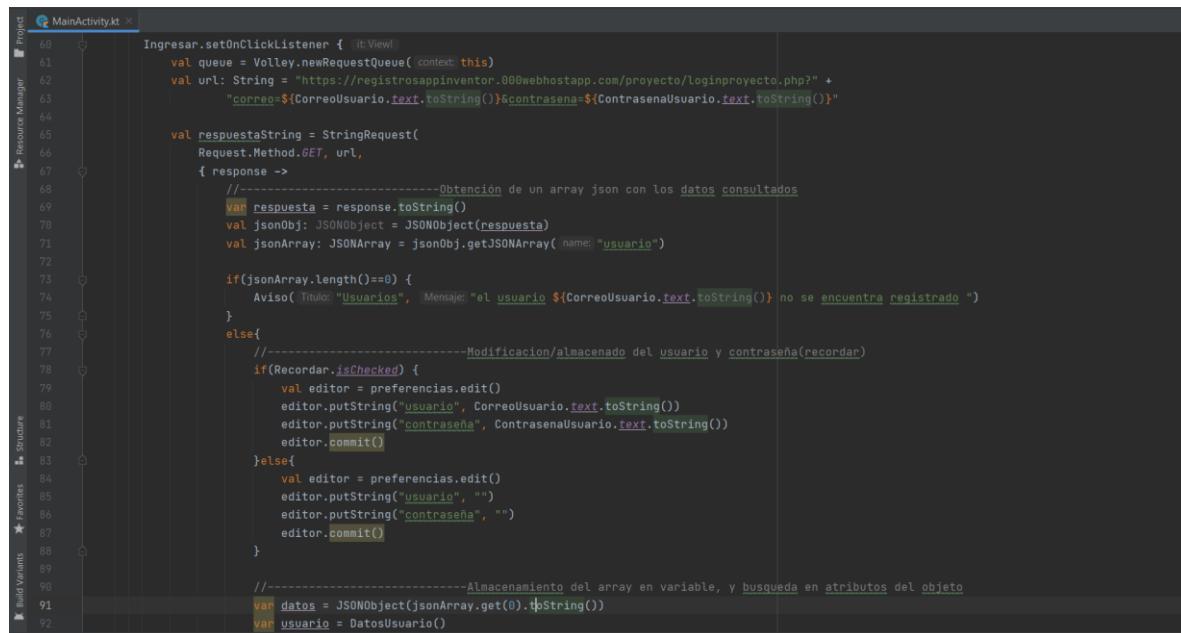
02/12/2021

Reporte:

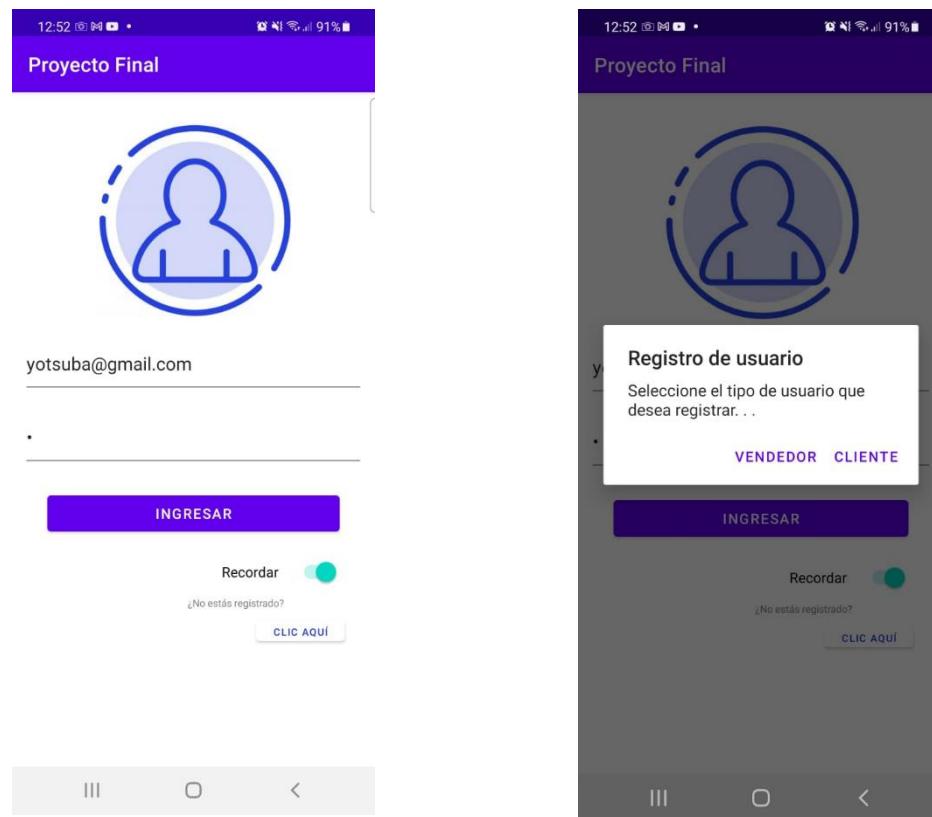
Archivo Main activity.....-----

Login:

Peticion http por medio de la clase volley a las apis del servidor del 00webhost, enviando el correo y usuario, para obtener un json de los datos obtenidos a raíz de la consulta.



```
MainActivity.kt
60 Ingresar.setOnClickListener { it: View ->
61     val queue = Volley.newRequestQueue(context: this)
62     val url: String = "https://registrosappinventor.00webhostapp.com/proyecto/loginprojeto.php?" +
63             "&correo=${CorreoUsuario.text.toString()}&contraseña=${ContraseñaUsuario.text.toString()}"
64
65     val respuestaString = StringRequest(
66         Request.Method.GET, url,
67         { response ->
68             //-----Obtención de un array json con los datos consultados
69             var respuesta = response.toString()
70             val jsonObj: JSONObject = JSONObject(respuesta)
71             val jsonArray: JSONArray = jsonObj.getJSONArray(name: "usuario")
72
73             if(jsonArray.length()==0) {
74                 Aviso( Título: "Usuarios", Mensaje: "el usuario ${CorreoUsuario.text.toString()} no se encuentra registrado ")
75             }
76             else{
77                 //-----Modificación/almacenado del usuario y contraseña(recordar)
78                 if(Recordar.isChecked) {
79                     val editor = preferencias.edit()
80                     editor.putString("usuario", CorreoUsuario.text.toString())
81                     editor.putString("contraseña", ContraseñaUsuario.text.toString())
82                     editor.commit()
83                 }
84                 else{
85                     val editor = preferencias.edit()
86                     editor.putString("usuario", "")
87                     editor.putString("contraseña", "")
88                     editor.commit()
89
90                 //-----Almacenamiento del array en variable, y búsqueda en atributos del objeto
91                 var datos = JSONObject(jsonArray.getJSONObject(0).toString())
92                 var usuario = DatosUsuario()
```



Registro:

Petición http para registrar el cliente

```
RegistroCliente.kt
56
57
58
59
60    fun registrarCliente(){
61        var queue = Volley.newRequestQueue( context: this)
62        val url: String = "https://registroappinventor.000webhostapp.com/proyecto/registro_usuarios.php?TipoUsuario=Cliente" +
63            "&NombreUsuario=${NombreCliente.text}&FotoUsuario=${urlImagenCliente.text}&MunicipioUsuario=${MunicipioCliente.selectedItem.toString()}" +
64            "&ColoniaUsuario=${ColoniaCliente.text}&DomicilioUsuario=${DomicilioCliente.text}&CorreoUsuario=${CorreoCliente.text}" +
65            "&ContrasenaUsuario=${ContrasenaCliente.text}&TelefonoUsuario=${TelefonoCliente.text}&VerificacionUsuario=${1}"
66
67        val stringReq = StringRequest(
68            Request.Method.GET, url,
69            { response ->
70                var strResp = response.toString()
71                if (strResp == "") {
72                    Aviso( Titulo: "Usuarios", Mensaje: "No se pudo registrar!")
73
74                } else { //mensaje de respuesta al registrar
75
76                    val dialogBuilder = AlertDialog.Builder( context: this)
77                    dialogBuilder.setMessage(strResp)
78                        .setCancelable(false)
79                        .setPositiveButton( text: "OK", DialogInterface.OnClickListener { dialog, id ->
80                            finish()
81                        })
82
83                    val alert = dialogBuilder.create()
84                    alert.setTitle("Registro de usuarios")
85                    alert.show()
86                }
87            },
88        )
89    }
90}
```

12:52 ⓘ ⓘ 91%

Nombre de usuario
username...

Foto de perfil

Url ...

Municipio Del Cliente
Colima

Colonia del cliente
Colonia

Domicilio del cliente
Domicilio #00

Correo del usuario
correo@ejemplo.com

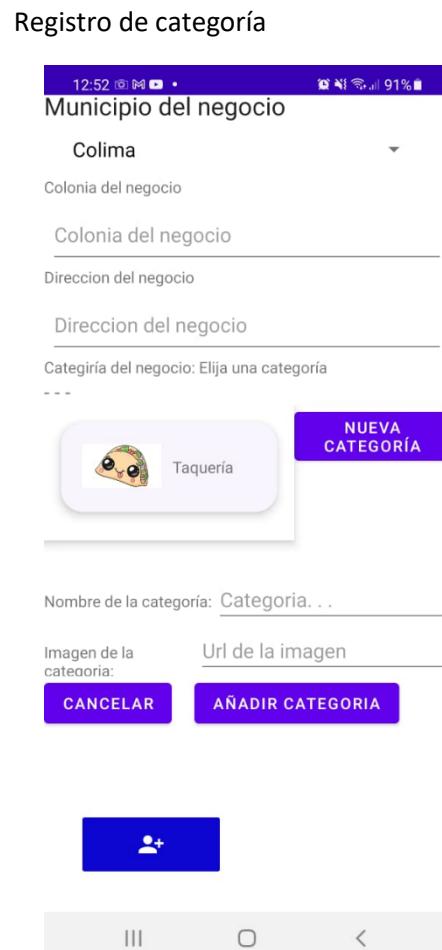
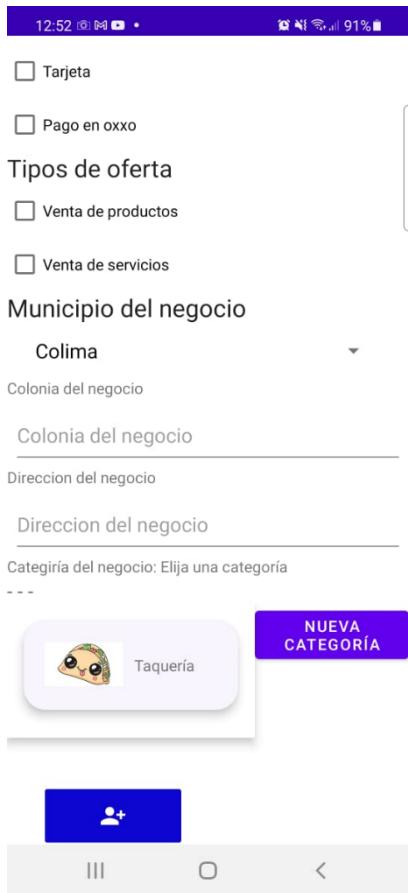
Contraseña

Teléfono
Teléfono

CANCELAR

☰ ⌂ <

Registro para vendedor: (se registra él y a un negocio)



Archivo “Principal”: (base de los elementos del menú):

Base del menú navegable:

Activity que funciona como base para el cambio de pantallas, en ella, al presionar un elemento del menú el fragment layout cambia.

The screenshot shows the Android Studio interface with two main panes. The top pane displays the `Principal.kt` file, which contains Java code for setting up a navigation menu and managing fragments. The bottom pane displays the `activity_main.xml` file, which defines the layout structure including a BottomNavigationView and a FrameLayout.

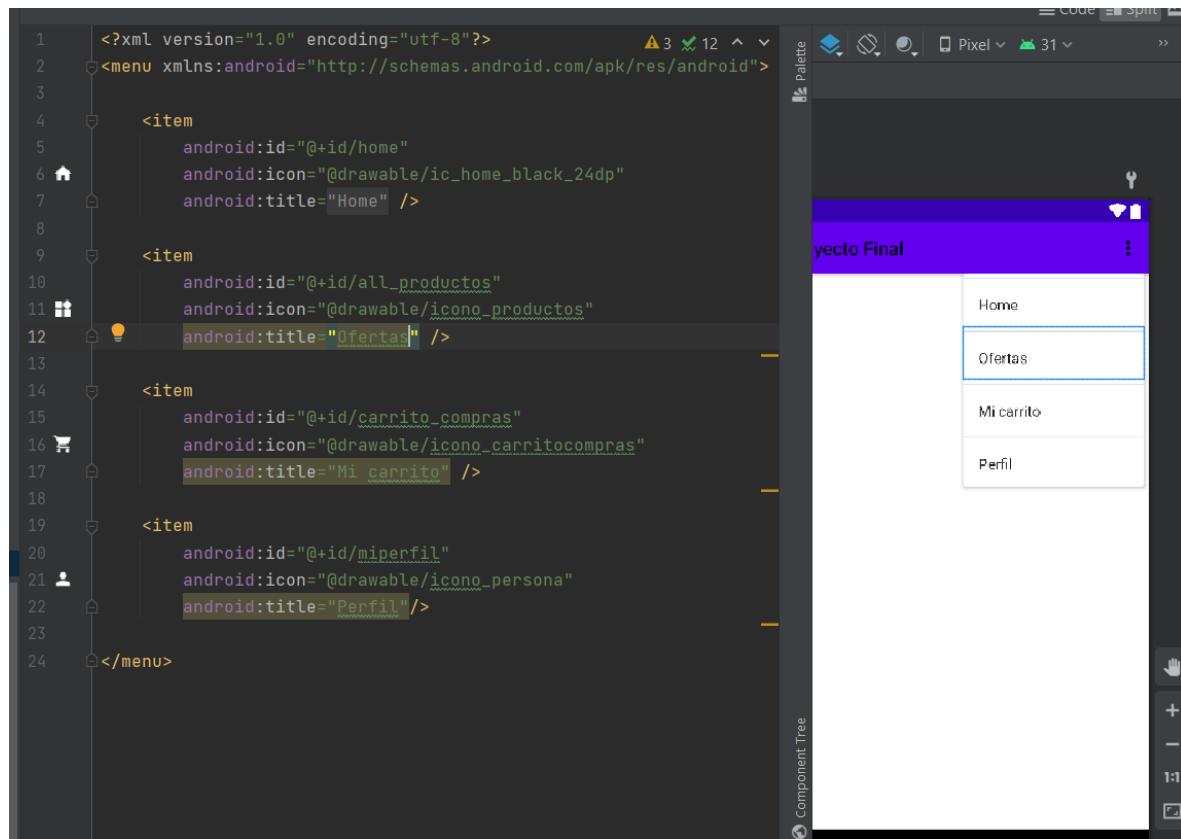
Principal.kt (Top Pane):

```
1 val perfil = Perfil.f()
2
3 val barra = supportActionBar
4 barra?.hide()
5
6 //-----Definición de home al iniciar sección
7 cambioFragment(miHome)
8
9 //----- Recibo de Datos
10 //----Datos del usuario
11 var bundle = intent.extras
12 var usuario = bundle?.getParcelable<DatosUsuario>( key: "datosUsuario")
13
14 //----- Envio de Datos
15 //envio del objeto datos usuario a fragment perfil
16 val infoPerfil = Bundle()
17 infoPerfil.putParcelable("infoPerfil", usuario)
18 perfil.arguments = infoPerfil
19 miHome.arguments = infoPerfil
20 carritoCompras.arguments = infoPerfil
21 productos.arguments = infoPerfil
22
23 //----- Cambio de fragments listener menu
24 menuNavegable.setOnNavigationItemSelectedListener { item: MenuItem?
25     when(item.itemId){
26         R.id.home -> cambioFragment(miHome)
27         R.id.all_productos -> cambioFragment(productos)
28         R.id.carrito_compras -> cambioFragment(carritoCompras)
29         R.id.miperfil -> cambioFragment(perfil)
30     }
31     true
32 }
```

activity_main.xml (Bottom Pane):

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".Principal">
8
9     <com.google.android.material.bottomnavigation.BottomNavigationView
10        android:id="@+id/menuNavegable"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        app:menu="@menu/menu"
14        app:layout_constraintBottom_toBottomOf="parent"
15        app:layout_constraintRight_toRightOf="parent"
16        app:layout_constraintLeft_toLeftOf="parent"/>
17
18     <FrameLayout
19        android:id="@+id/contenedor"
20        android:layout_width="match_parent"
21        android:layout_height="match_parent"
22        app:layout_constraintBottom_toBottomOf="parent"
23        app:layout_constraintRight_toRightOf="parent" />
24
25 </androidx.constraintlayout.widget.ConstraintLayout>
```

Menú:



Home:

Consulta de datos; Negocios:

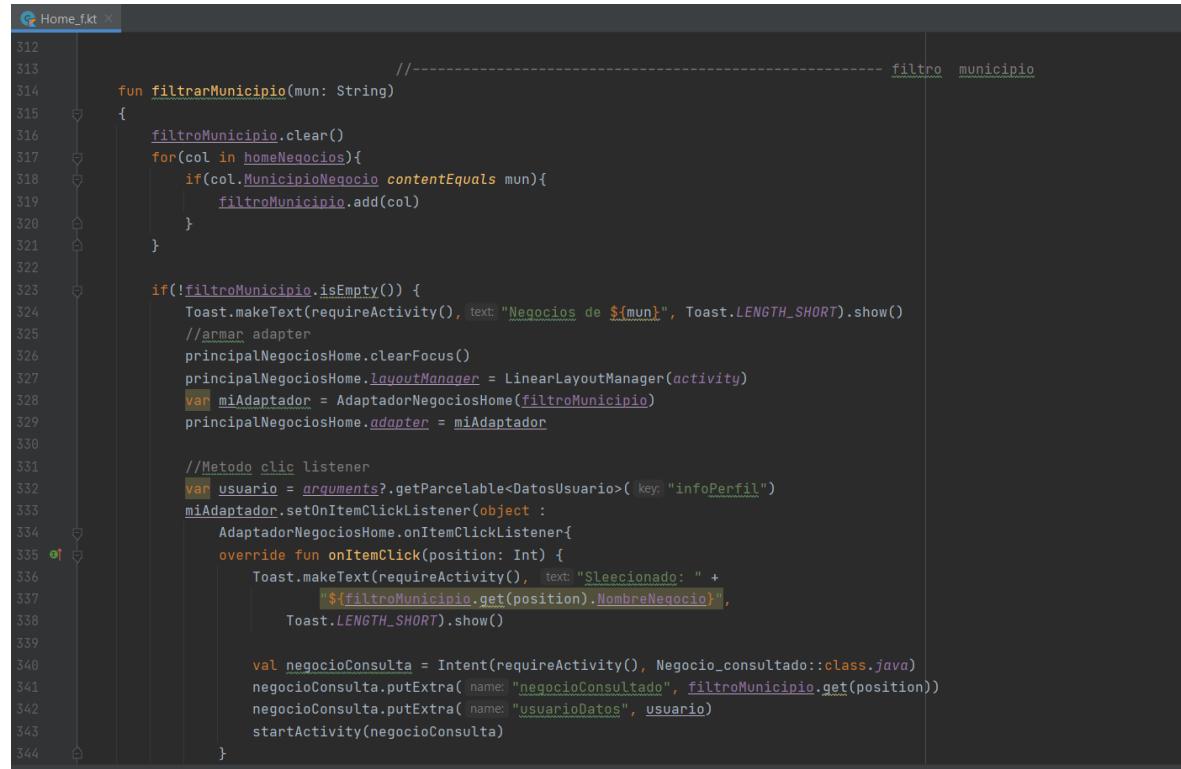
```
fun listarHome(usuario: DatosUsuario?) {
    val queue = Volley.newRequestQueue(activity)
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/listar.php?comando=NegociosHome"

    val respuestaString = StringRequest(
        Request.Method.GET, url, { response ->
            //-----Obtención de un array json con los datos consultados
            var respuesta = response.toString()
            val jsonObj: JSONObject = JSONObject(respuesta)
            val jsonArray: JSONArray = jsonObj.getJSONArray( name: "miHome")
            if (jsonArray.length() == 0) {
                Aviso( Titulo: "Aviso", Mensaje: "no se encuentran negocios")
            } else {
                homeNegocios.clear()
                var indice = 0
                //-----Almacenamiento del array en variable, y busqueda en atributos del objeto.clear()
                for (i in 0 until jsonArray.length()) {
                    var elNegocio = DatosNegocio()
                    var datos = JSONObject(jsonArray.get(indice).toString())

                    elNegocio.idnegocio = datos.getString( name: "idnegocio").toString()
                    elNegocio.CorreoUsuario = datos.getString( name: "CorreoUsuario").toString()
                    elNegocio.NombreNegocio = datos.getString( name: "NombreNegocio").toString()
                    elNegocio.LogoNegocio = datos.getString( name: "LogoNegocio").toString()
                    elNegocio.Efectivo = datos.getString( name: "Efectivo").toString()
                    elNegocio.Tarjeta = datos.getString( name: "Tarjeta").toString()
                    elNegocio.PagoOxxo = datos.getString( name: "PagoOxxo").toString()
                    elNegocio.Oferta_producto = datos.getString( name: "Oferta_producto").toString()
                    elNegocio.Oferta_servicio = datos.getString( name: "Oferta_servicio").toString()
                }
            }
        }
    )
}
```

Consulta por municipio:

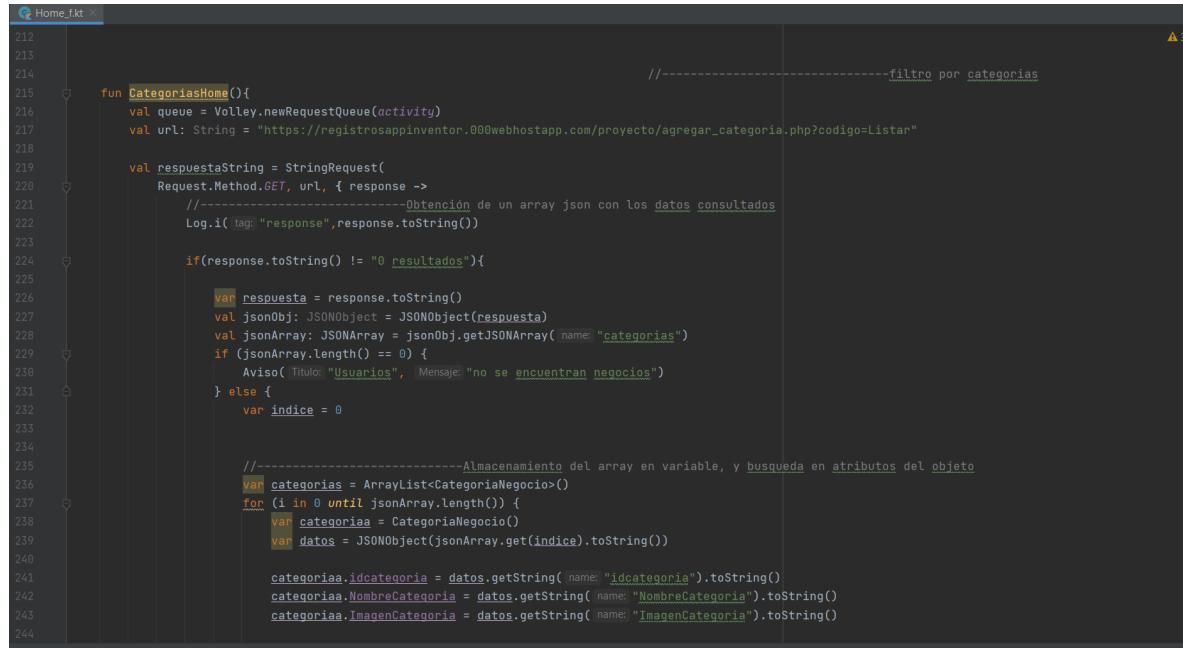
Llena una lista de objetos negocio llamada filtro municipio, esta lista es usada para rellenar el recycler



```
312
313
314     fun filtrarMunicipio(mun: String) {
315         filtroMunicipio.clear()
316         for(col in homeNegocios){
317             if(col.MunicipioNegocio contentEquals mun){
318                 filtroMunicipio.add(col)
319             }
320         }
321     }
322
323     if(!filtroMunicipio.isEmpty()) {
324         Toast.makeText(requireActivity(), text: "Negocios de ${mun}", Toast.LENGTH_SHORT).show()
325         //armar adapter
326         principalNegociosHome.clearFocus()
327         principalNegociosHome.layoutManager = LinearLayoutManager(activity)
328         var miAdaptador = AdaptadorNegociosHome(filtroMunicipio)
329         principalNegociosHome.adapter = miAdaptador
330
331         //Metodo clic listener
332         var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil")
333         miAdaptador.setOnItemClickListener(object :
334             AdaptadorNegociosHome.OnItemClickListener{
335                 override fun onItemClick(position: Int) {
336                     Toast.makeText(requireActivity(), text: "Sleecionado: " +
337                         "${filtroMunicipio.get(position).NombreNegocio}",
338                         Toast.LENGTH_SHORT).show()
339
340                     val negocioConsulta = Intent(requireActivity(), Negocio_consultado::class.java)
341                     negocioConsulta.putExtra( name: "negocioConsultado", filtroMunicipio.get(position))
342                     negocioConsulta.putExtra( name: "usuarioDatos", usuario)
343                     startActivity(negocioConsulta)
344                 }
345             })
```

Filtro por categorías:

Se listan las categorías existentes en un recyclerview, y se agrega el método clic listener, dónde se agrega al recycler de los negocios, el nombre de la categoría, lo cual hace que se filtren los negocios por categoría.



The screenshot shows the Java code for a RecyclerView adapter named Home.fkt. The code is used to fetch categories from a JSON API and store them in a list of objects. It includes annotations for code review, such as 'filtro por categorías' and 'Almacenamiento del array en variable, y búsqueda en atributos del objeto'. The code uses Volley for network requests and JSONObject/JSONArray for parsing JSON data.

```
212  
213  
214  
215     fun CategoriasHome(){  
216         val queue = Volley.newRequestQueue(activity)  
217         val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/agregar_categoria.php?codigo=Listar"  
218  
219         val respuestaString = StringRequest(  
220             Request.Method.GET, url, { response ->  
221                 //-----  
222                 //Obtención de un array json con los datos consultados  
223                 Log.i( tag: "response", response.toString())  
224  
225                 if(response.toString() != "0 resultados"){  
226                     var respuesta = response.toString()  
227                     val jsonObj: JSONObject = JSONObject(respuesta)  
228                     val jsonArray: JSONArray = jsonObj.getJSONArray( name: "categorias")  
229                     if (jsonArray.length() == 0) {  
230                         Aviso( Titulo: "Usuarios", Mensaje: "no se encuentran negocios")  
231                     } else {  
232                         var indice = 0  
233  
234  
235                         //-----  
236                         //Almacenamiento del array en variable, y búsqueda en atributos del objeto  
237                         var categorias = ArrayList<CategoriaNegocio>()  
238                         for (i in 0 until jsonArray.length()) {  
239                             var categoriaa = CategoriaNegocio()  
240                             var datos = JSONObject(jsonArray.get(indice).toString())  
241  
242                             categoriaa.idcategoria = datos.getString( name: "idcategoria").toString()  
243                             categoriaa.NombreCategoria = datos.getString( name: "NombreCategoria").toString()  
244                             categoriaa.ImagenCategoria = datos.getString( name: "ImagenCategoria").toString()
```

Buscador:

A tiempo real, crea una consulta donde carácter por carácter escrito, va filtrando los elementos de la lista, dando la posibilidad de buscar el nombre del negocio, o buscar las colonias de los negocios y de esta manera evitar la rigidez de la consulta, ya que se ignoran mayúsculas, minúsculas y acentos o tildes.

```
//----- toda esta jalada es el algoritmo de busqueda
override fun onQueryTextChange(newText: String?): Boolean {
    if (newText!!.isNotEmpty()){
        busquedaNegocio.clear()
        val busqueda = newText.toLowerCase(Locale.getDefault())
        homeNegocios.forEach { it.DatosNegocio
            if (it.NombreNegocio.toLowerCase(Locale.getDefault()).contains(busqueda) ||
                it.ColoniasNegocio.toLowerCase(Locale.getDefault()).contains(busqueda)) {
                busquedaNegocio.add(it)
            }
        }
        //armar adapter
        principalNegociosHome.clearFocus()
        principalNegociosHome.layoutManager = LinearLayoutManager(activity)
        var miAdaptador = AdaptadorNegociosHome(busquedaNegocio)
        principalNegociosHome.adapter = miAdaptador

        //Metodo clic listener
        miAdaptador.setOnItemClickListener(object :
            AdaptadorNegociosHome.OnItemClickListener{
            override fun onItemClick(position: Int) {
                Toast.makeText(requireActivity(), text: "Slecciónado: " +
                    "${busquedaNegocio.get(position).NombreNegocio}",
                    Toast.LENGTH_SHORT).show()

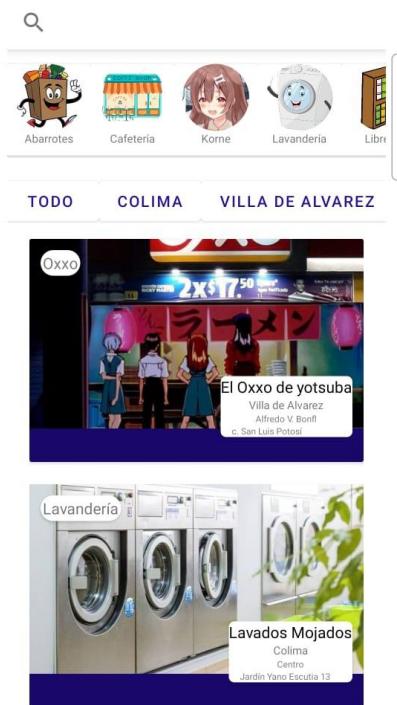
                val negocioConsulta = Intent(requireActivity(), Negocio_consultado::class.java)
                negocioConsulta.putExtra( name: "negocioConsultado", busquedaNegocio.get(position))
                negocioConsulta.putExtra( name: "usuarioDatos", usuario)
                startActivity(negocioConsulta)
            }
        })
    }
}
```

```
} else{
    //armar adapter
    principalNegociosHome.clearFocus()
    principalNegociosHome.layoutManager = LinearLayoutManager(activity)
    var miAdaptador = AdaptadorNegociosHome(homeNegocios)
    principalNegociosHome.adapter = miAdaptador

    //Metodo clic listener
    miAdaptador.setOnItemClickListener(object :
        AdaptadorNegociosHome.OnItemClickListener{
        override fun onItemClick(position: Int) {
            Toast.makeText(requireActivity(), text: "Slecciónado: " +
                "${homeNegocios.get(position).NombreNegocio}",
                Toast.LENGTH_SHORT).show()

            val negocioConsulta = Intent(requireActivity(), Negocio_consultado::class.java)
            negocioConsulta.putExtra( name: "negocioConsultado", filtroCategoria.get(position))
            negocioConsulta.putExtra( name: "usuarioDatos", usuario)
            startActivity(negocioConsulta)
        }
    })
}
return true
}
```

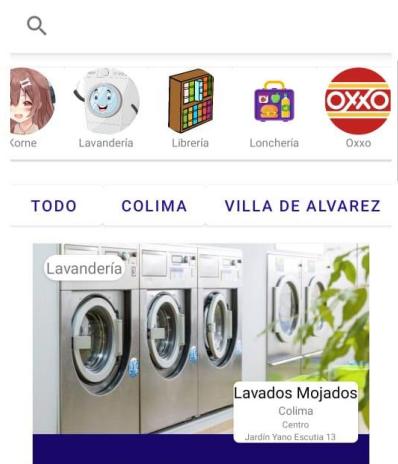
12:52 ☰ 🔍 91%



Home

Home(Negocios ordenados
alfabeticamente)

1:29 ☰ 🔍 89%



Home

filtro por categoría

1:29 ☰ 🔍 89%

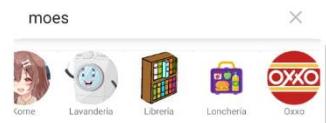


Negocios de Villa de Alvarez

Home

filtro por municipio

1:30 ☰ 🔍 89%

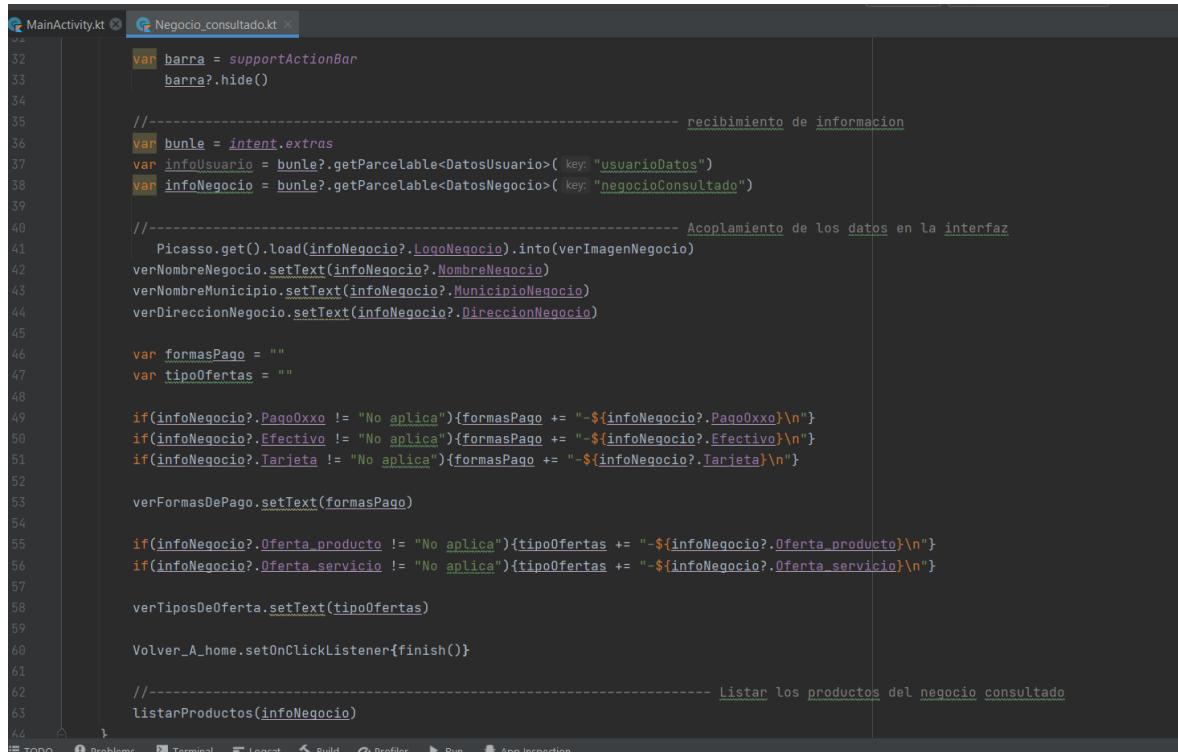


Home

filtro por búsqueda

Negocio consultado: ver los productos de un negocio en específico:

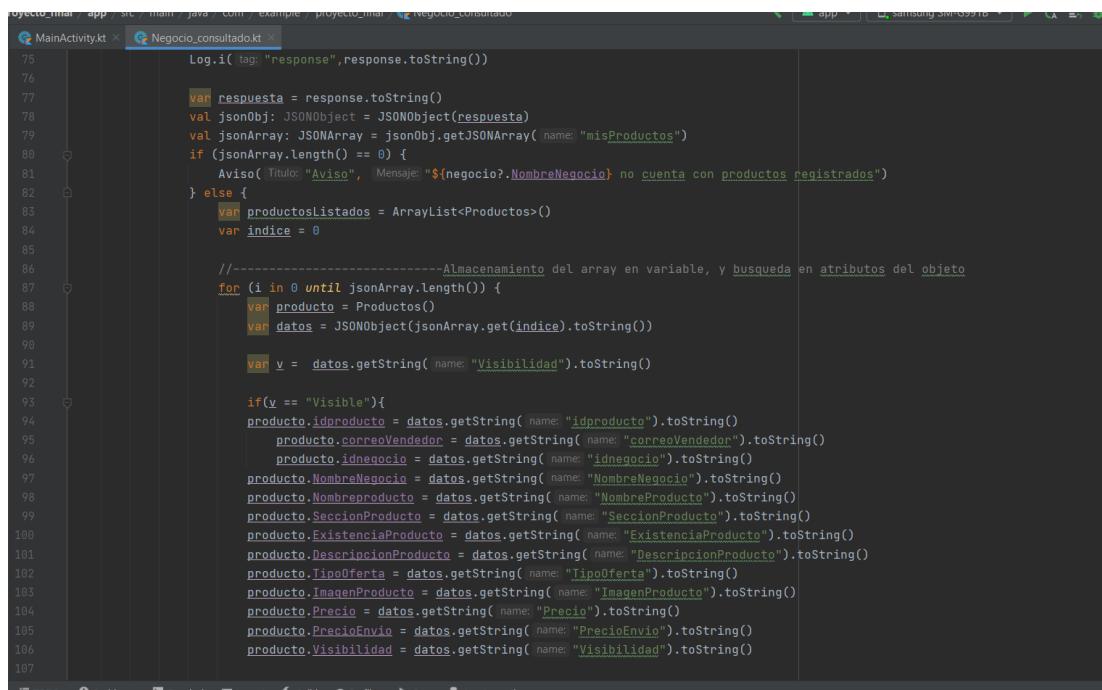
Mandar los datos del elemento clickeado al activity de negocio consultado.



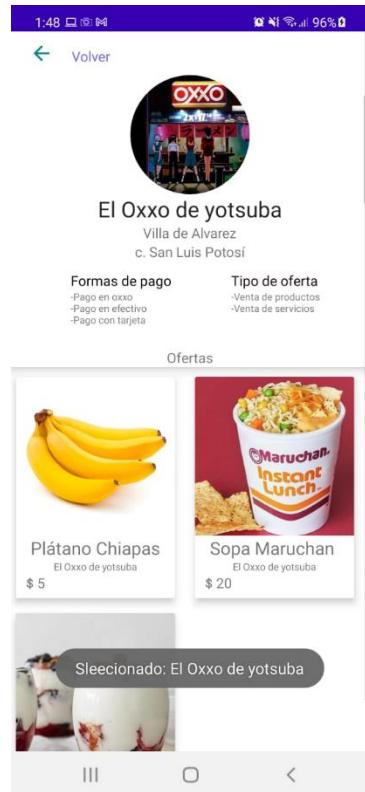
```
32 var barra = supportActionBar
33 barra?.hide()
34
35 //----- recibimiento de informacion
36 var bunle = intent.extras
37 var infoUsuario = bunle?.getParcelable<DatosUsuario>( key: "usuarioDatos")
38 var infoNegocio = bunle?.getParcelable<DatosNegocio>( key: "negocioConsultado")
39
40 //----- Acoplamiento de los datos en la interfaz
41 Picasso.get().load(infoNegocio?.LogoNegocio).into(verImagenNegocio)
42 verNombreNegocio.setText(infoNegocio?.NombreNegocio)
43 verNombreMunicipio.setText(infoNegocio?.MunicipioNegocio)
44 verDireccionNegocio.setText(infoNegocio?.DireccionNegocio)
45
46 var formasPago = ""
47 var tipoOfertas = ""
48
49 if(infoNegocio?.PagoOxxo != "No aplica"){formasPago += "-${infoNegocio?.PagoOxxo}\n"}
50 if(infoNegocio?.Efectivo != "No aplica"){formasPago += "-${infoNegocio?.Efectivo}\n"}
51 if(infoNegocio?.Tarjeta != "No aplica"){formasPago += "-${infoNegocio?.Tarjeta}\n"}
52
53 verFormasDePago.setText(formasPago)
54
55 if(infoNegocio?.Oferta_producto != "No aplica"){tipoOfertas += "-${infoNegocio?.Oferta_producto}\n"}
56 if(infoNegocio?.Oferta_servicio != "No aplica"){tipoOfertas += "-${infoNegocio?.Oferta_servicio}\n"}
57
58 verTiposDeOferta.setText(tipoOfertas)
59
60 Volver_A_home.setOnClickListener{finish()}
61
62 //----- Listar los productos del negocio consultado
63 listarProductos(infoNegocio)
64
```

Listar productos del negocio consultado:

Consulta de los productos del negocio.



```
75 Log.i( tag: "response", response.toString())
76
77 var respuesta = response.toString()
78 val jsonObj: JSONObject = JSONObject(respuesta)
79 val jsonArray: JSONArray = jsonObj.getJSONArray( name: "misProductos")
80 if (jsonArray.length() == 0) {
81     Aviso( Titulo: "Aviso", Mensaje: "${negocio?.NombreNegocio} no cuenta con productos registrados")
82 } else {
83     var productosListados = ArrayList<Productos>()
84     var indice = 0
85
86     //-----Almacenamiento del array en variable, y busqueda en atributos del objeto
87     for (i in 0 until jsonArray.length()) {
88         var producto = Productos()
89         var datos = JSONObject(jsonArray.get(indice).toString())
90
91         var v =  datos.getString( name: "Visibilidad").toString()
92
93         if(v == "Visible"){
94             producto.idproducto = datos.getString( name: "idproducto").toString()
95             producto.correoVendedor = datos.getString( name: "correoVendedor").toString()
96             producto.idnegocio = datos.getString( name: "idnegocio").toString()
97             producto.NombreNegocio = datos.getString( name: "NombreNegocio").toString()
98             producto.NombreProducto = datos.getString( name: "NombreProducto").toString()
99             producto.SeccionProducto = datos.getString( name: "SeccionProducto").toString()
100            producto.ExistenciaProducto = datos.getString( name: "ExistenciaProducto").toString()
101            producto.DescripcionProducto = datos.getString( name: "DescripcionProducto").toString()
102            producto.TipoOferta = datos.getString( name: "TipoOferta").toString()
103            producto.ImagenProducto = datos.getString( name: "ImagenProducto").toString()
104            producto.Precio = datos.getString( name: "Precio").toString()
105            producto.PrecioEnvio = datos.getString( name: "PrecioEnvio").toString()
106            producto.Visibilidad = datos.getString( name: "Visibilidad").toString()
107        }
108    }
109
110    var lista = ArrayAdapter<Productos>(this, android.R.layout.simple_list_item_1, productosListados)
111    lista.notifyDataSetChanged()
112    lista.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
113    spinner.adapter = lista
114
115    spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
116        override fun onItemSelected(parent: AdapterView<*>, view: View, position: Int, id: Long) {
117            var producto = spinner.getItemAtPosition(position) as Productos
118            var nombre = producto.NombreNegocio + " - " + producto.NombreProducto
119            var precio = producto.Precio
120            var cantidad = producto.ExistenciaProducto
121            var tipo = producto.TipoOferta
122            var imagen = producto.ImagenProducto
123
124            var intento = Intent(this@Negocio_consultado, DetalleActivity::class.java)
125            intento.putExtra("nombre", nombre)
126            intento.putExtra("precio", precio)
127            intento.putExtra("cantidad", cantidad)
128            intento.putExtra("tipo", tipo)
129            intento.putExtra("imagen", imagen)
130            startActivity(intento)
131        }
132
133        override fun onNothingSelected(parent: AdapterView<*>){}
134    }
135
136    spinner.setOnItemClickListener { parent, view, position, id ->
137        var producto = spinner.getItemAtPosition(position) as Productos
138        var nombre = producto.NombreNegocio + " - " + producto.NombreProducto
139        var precio = producto.Precio
140        var cantidad = producto.ExistenciaProducto
141        var tipo = producto.TipoOferta
142        var imagen = producto.ImagenProducto
143
144        var intento = Intent(this@Negocio_consultado, DetalleActivity::class.java)
145        intento.putExtra("nombre", nombre)
146        intento.putExtra("precio", precio)
147        intento.putExtra("cantidad", cantidad)
148        intento.putExtra("tipo", tipo)
149        intento.putExtra("imagen", imagen)
150        startActivity(intento)
151    }
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2
```



Todas las ofertas:

Es una lista de productos y servicios que existan en la base de datos, ordenados del más antiguo al más reciente.

```
62
63     fun listarProductos(){
64
65         val queue = Volley.newRequestQueue(activity)
66         val url: String = "https://registrosappinvendor.000webhostapp.com/proyecto/listar.php?comando=ListarTodo"
67
68         val respuestaString = StringRequest(
69             Request.Method.GET, url, { response ->
70                 //-----Obtención de un array json con los datos consultados
71                 Log.i( tag: "response", response.toString())
72
73                 var respuesta = response.toString()
74                 val jsonObj: JSONObject = JSONObject(respuesta)
75                 val jsonArray: JSONArray = jsonObj.getJSONArray( name: "todasOfertas")
76                 if (jsonArray.length() == 0) {
77                     Aviso( Titulo: "Aviso", Mensaje: "no se encontraron productos registrados")
78                 } else {
79                     todosLosProductos.clear()
80                     var indice = 0
81
82                     //-----Almacenamiento del array en variable, y búsqueda en atributos del objeto
83                     for (i in 0 until jsonArray.length()) {
84                         var producto = Productos()
85                         var datos = JSONObject(jsonArray.get(indice).toString())
86
87                         var v =  datos.getString( name: "Visibilidad").toString()
88
89                         if(v == "Visible"){
90                             producto.idproducto = datos.getString( name: "idproducto").toString()
91                             producto.correoVendedor = datos.getString( name: "correoVendedor").toString()
92                             producto.idnegocio = datos.getString( name: "idnegocio").toString()
93                             producto.NombreNegocio = datos.getString( name: "NombreNegocio").toString()
94                             producto.Nombreproducto = datos.getString( name: "Nombreproducto").toString()
95                             producto.CantidadProducto = datos.getInt( name: "CantidadProducto").toString()
96                         }
97                     }
98                 }
99             }
100            todosLosProductos.notifyDataSetChanged()
101        }
102    }
103
104    
```

Buscador de las ofertas: buscar por nombre del producto, nombre del negocio, o sección del producto.

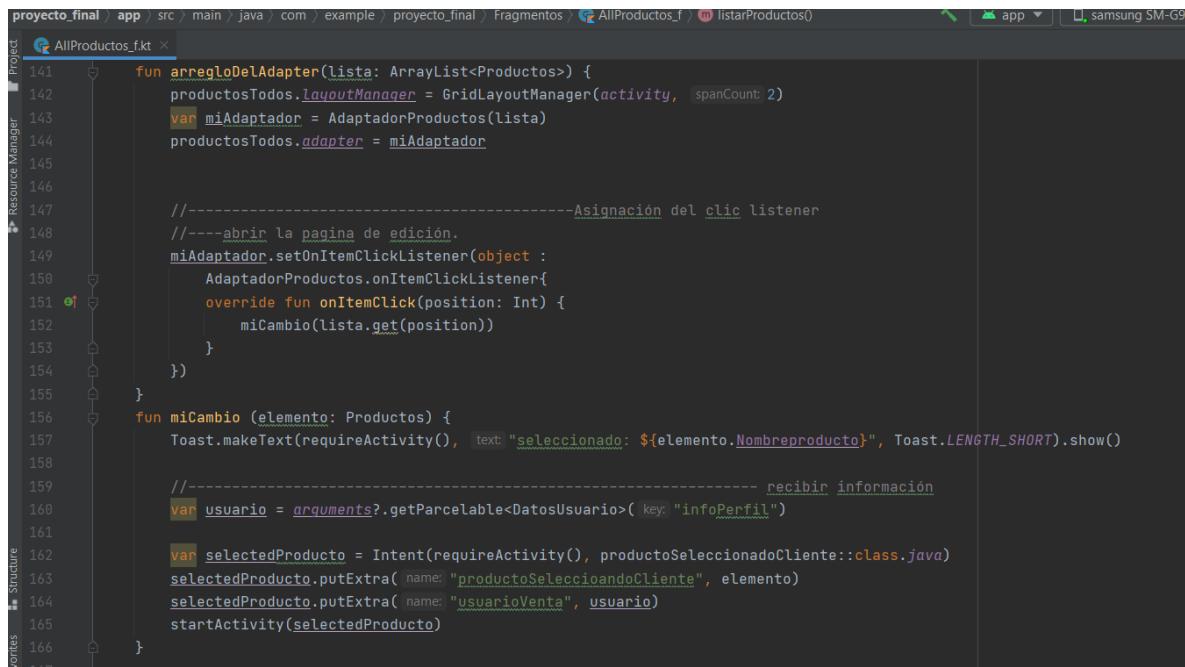
```
//----- listar
listarProductos()

//-----buscar
buscadorProductos.setOnQueryTextListener(object:
    androidx.appcompat.widget.SearchView.OnQueryTextListener{
    override fun onQueryTextSubmit(query: String?): Boolean {
        return true
    }

    override fun onQueryTextChange(newText: String?): Boolean {
        if(newText!!.isNotEmpty()){
            productosBuscados.clear()
            val busqueda = newText.toLowerCase(Locale.getDefault())

            todosLosProductos.forEach{ it: Productos
                if(it.Nombreproducto.toLowerCase(Locale.getDefault()).contains(busqueda)|||it.NombreNegocio.toLowerCase(Locale.getDefault()).contains(busqueda) ||| it.SeccionProducto.toLowerCase(Locale.getDefault()).contains(busqueda)){
                    productosBuscados.add(it)
                }
            }
            arregloDelAdapter(productosBuscados)
        }else {arregloDelAdapter(todosLosProductos)}
        return true
    }
})
```

Función del buscador donde se rearma el recycler por carácter consultado:



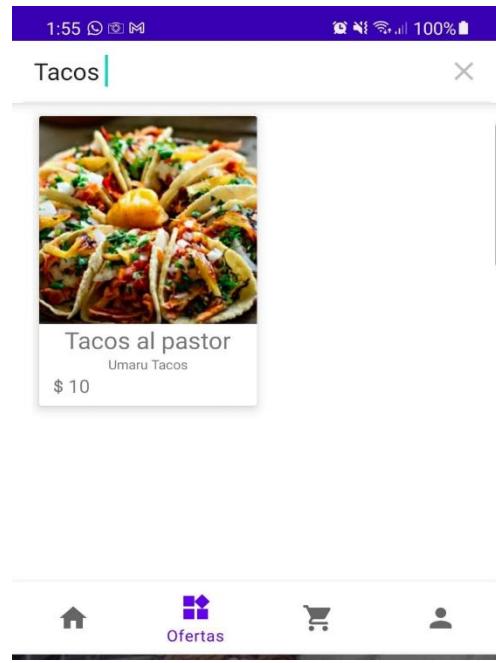
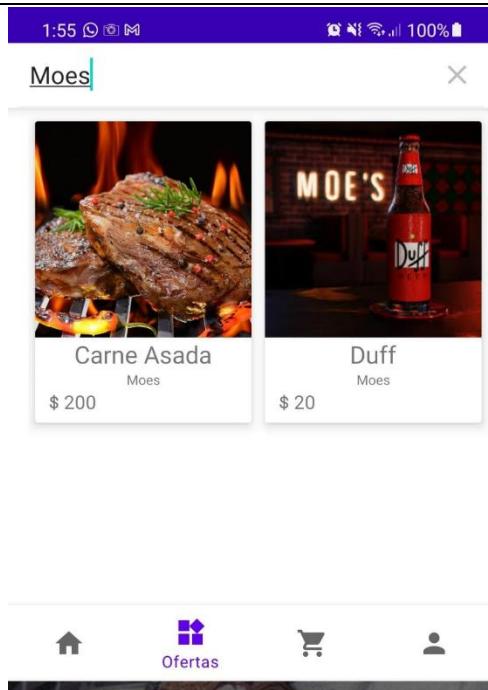
```
funcion arregloDelAdapter(lista: ArrayList<Productos>){
    productosTodos.layoutManager = GridLayoutManager(activity, spanCount: 2)
    var miAdaptador = AdaptadorProductos(lista)
    productosTodos.adapter = miAdaptador

    //-----Asignación del clic listener
    //----abrir la pagina de edición.
    miAdaptador.setOnItemClickListener(object : AdaptadorProductos.onItemClickListener{
        override fun onItemClick(position: Int) {
            miCambio(lista.get(position))
        }
    })
}

func miCambio (elemento: Productos) {
    Toast.makeText(requireActivity(), text: "seleccionado: ${elemento.Nombreproducto}", Toast.LENGTH_SHORT).show()

    //----- recibir información
    var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil")

    var selectedProducto = Intent(requireActivity(), productoSeleccionadoCliente::class.java)
    selectedProducto.putExtra( name: "productoSeleccionandoCliente", elemento)
    selectedProducto.putExtra( name: "usuarioVenta", usuario)
    startActivityForResult(selectedProducto)
}
```



Detalles del los productos:

Al dar clic a un producto, ya sea en la pagina del negocio, en el feed, consulta o carrito de ventas, envía a la pagina de detalles del producto, donde se encuentran los detalles del producto como cantidad, precio, promedio de votaciones (estrellas), comentarios y respuestas de estos:

Clic compras y cantidades.

```
app / src / main / java / com / example / proyecto_final / productoSeleccionadoCliente.kt ConfirmarCompra(usuario: DatosUsuario?, producto: Producto?) {
    var cantidadActualizable = cantidadActualizable
    var precioFinal = (cantidad * precio!!).toFloat()
    consultaCPrecio.setText("$ " + precioFinal)
    consultaCEExistencias.setText(cantidadActualizable.toString())
}

incremento.setOnClickListener { it: View! ->
    if(cantidad < existencias!!){
        cantidad++
        cantidadActualizable--
        CantidadAVender.setText(cantidad.toString())
        precioFinal = (cantidad * precio!!).toFloat()
        consultaCPrecio.setText("$ " + precioFinal)
        consultaCEExistencias.setText(cantidadActualizable.toString())
    }
}

comprarProductoServicio.setOnClickListener { it: View! ->
    if(elProducto?.TipoOferta != "Venta de un servicio"){
        if(CantidadAVender.text.isNotEmpty() && CantidadAVender.text.toString() != "0"){
            //asignar el precio final
            var cantidadUsuario = CantidadAVender.text.toInt()
            var pE = (precio!! * cantidadUsuario)

            if(cantidadUsuario < existencias!!) {
                //modificar las existencias y tratarlas como la cantidad compradas
                elProducto?.ExistenciaProducto = cantidadUsuario.toString()
                elProducto?.Precio = (pE + precioEnvio!!).toString()
                //disminuir existencias y al comprar... actualizar esa reducción.
                var a = (cantidadActualizable - cantidadUsuario).toInt()
                ConfirmarCompra(elUsuario, elProducto, a)
            }else{
                Toast.makeText(context, "La cantidad seleccionada no coincide con la cantidad disponible.", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Mandar al carro:

```
        if(cajaComentario.text.isNotEmpty()) escribirComentario(@ElProducto, @ElUsuario, comando: "comentar")
    }

fun mandaAlCarro(usuario: DatosUsuario?, producto: Productos?, cantidadSolicitada:Int){
    val queue = Volley.newRequestQueue( context: this)
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/carrito_compras.php?comando=agregarAlCarro"+
        "&correoCliente=${usuario?.CorreoUsuario}&idproducto=${producto?.idproducto}&CantidadSolicitada=${cantidadSolicitada}" +
        "&idnegocio=${producto?.idnegocio}&correoVendedor=${producto?.correoVendedor}&NombreNegocio=${producto?.NombreNegocio}&NombreProducto" +
        "&SeccionProducto=${producto?.SeccionProducto}&ExistenciaProducto=${consultaCExistencias.text}" +
        "&TipoOferta=${producto?.TipoOferta}&ImagenProducto=${producto?.ImagenProducto}&Precio=${producto?.Precio}" +
        "&PrecioEnvio=${producto?.PrecioEnvio}&DescripcionProducto=${producto?.DescripcionProducto}"
}

val stringReq = StringRequest(
    Request.Method.GET, url,
    { response ->
        var strResp = response.toString()
        if (strResp == "") {
            Aviso( Titulo: "Producto", Mensaje: "No se pudo registrar!")
        } else { //mensaje de respuesta al registrar
            val dialogBuilder = AlertDialog.Builder( context: this)
            dialogBuilder.setMessage(strResp)
                .setCancelable(false)
                .setPositiveButton( text: "OK", DialogInterface.OnClickListener { dialog, id ->
                    dialog.cancel()
                })
            val alert = dialogBuilder.create()
            alert.setTitle("Registro de Productos")
            alert.show()
        }
    }
)
```

Comentarios (preguntas al vendedor y respuestas de este mismo).

Listar comentarios:

```
AllProductos.fkt ✘ productoSeleccionadoCliente.kt ✘
```

```
        }

fun ListarComentarios(productos: Productos?, usuario: DatosUsuario?) {
    val queue = Volley.newRequestQueue( context: this)
    val url: String =
        "https://registrosappinventor.000webhostapp.com/proyecto/comentarios.php?comando=ListarComentarios"+
        "&idproducto=${productos?.idproducto}"

    val respuestaString = StringRequest(
        Request.Method.GET, url, { response -
            //-----Obtención de un array json con los datos consultados
            Log.i( tag: "response", response.toString())

            var respuesta = response.toString()
            val jsonObj: JSONObject = JSONObject(respuesta)
            val jsonArray: JSONArray = jsonObj.getJSONArray( name: "comentarios")
            if (jsonArray.length() > 0){
                var indice = 0
                //-----Almacenamiento del array en variable, y búsqueda en atributos del objeto
                var comentarioProducto= ArrayList<DatosComentario>()

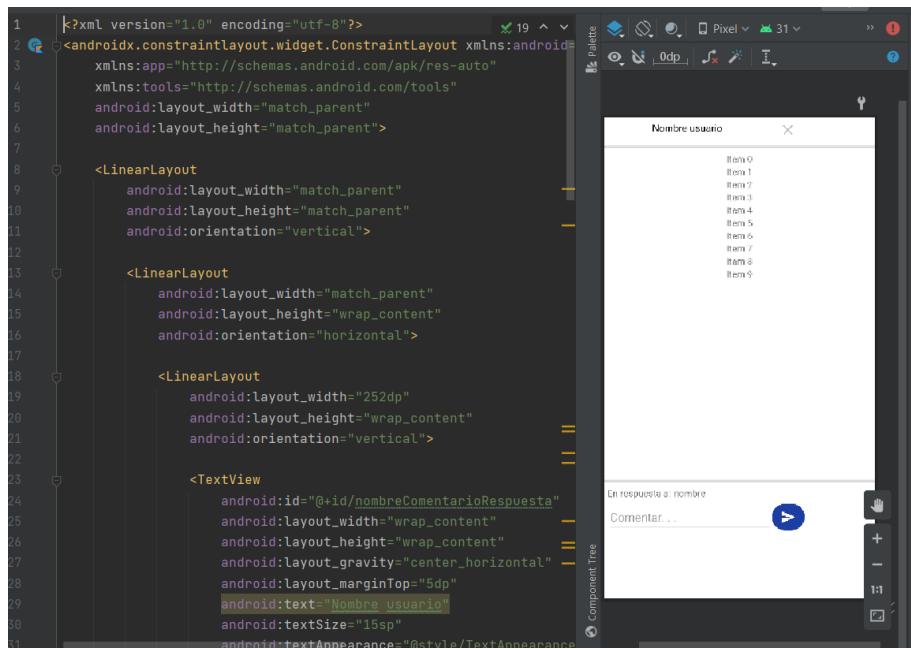
                for (i in 0 until jsonArray.length()) {
                    var coment = DatosComentario()
                    var datos = JSONObject(jsonArray.get(indice).toString())
                    coment.id_comentario = datos.getString( name: "idcomentario").toString()
                    coment.id_comentario_respuesta = datos.getString( name: "idcomentariorespuesta").toString()
                    coment.id_producto = datos.getString( name: "idproducto").toString()
                    coment.Tipo_Comentario = datos.getString( name: "TipoComentario").toString()
                    coment.Nombre_Usuario= datos.getString( name: "NombreUsuario").toString()
                    coment.Nombre_Usuario_Respuesta = datos.getString( name: "NombreUsuarioRespuesta").toString()
                    coment.Imagen_Usuario = datos.getString( name: "ImagenUsuario").toString()
                    coment.Contenido = datos.getString( name: "Contenido").toString()
                    coment.Fecha_Comentario = datos.getString( name: "FechaComentario").toString()
                }
            }
        }
    )
}
```

Escribir comentarios:

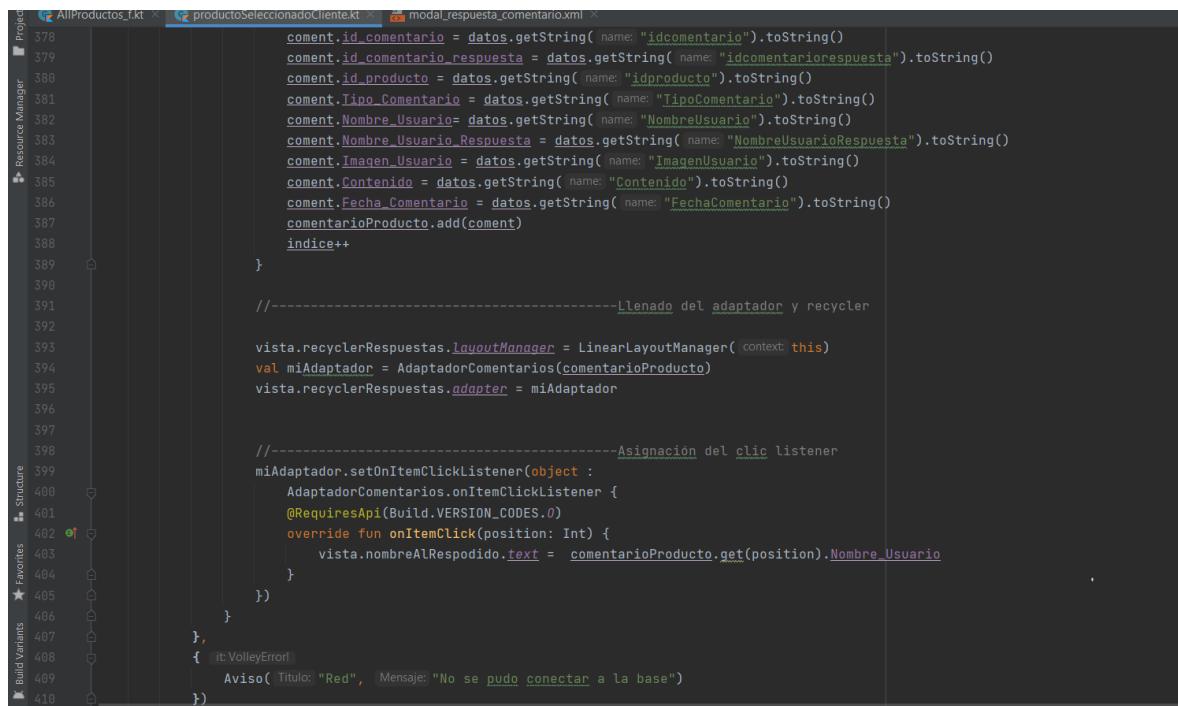
```
1 @RequiresApi(Build.VERSION_CODES.O)
2 fun EscribirComentario(productos: Productos?, usuario: DatosUsuario?, comando:String){
3     var dtf = DateTimeFormatter.ofPattern(pattern: "dd/MM/yyyy HH:mm:ss")
4     var fecha = LocalDateTime.now()
5     var f = dtf.format(fecha).toString()
6
7     var queue = Volley.newRequestQueue(context: this)
8     val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/comentarios.php?comando=comentar"+
9         "&idproducto=${productos?.idproducto}&TipoComentario=Comentario&NombreUsuario=${usuario?.NombreUsuario}"+
10        "&ImagenUsuario=${usuario?.FotoUsuario}&Contenido=${cajaComentario.text}&FechaComentario=${f}"
11
12     val stringReq = StringRequest(
13         Request.Method.GET, url,
14         { response ->
15             var strResp = response.toString()
16             if (strResp == "") {
17                 Aviso(Titulo: "Usuarios", Mensaje: "No se pudo registrar!")
18
19             } else { //mensaje de respuesta al registrar
20                 cajaComentario.setText("")
21                 ListarComentarios(productos, usuario)
22             }
23         },
24         { it: VolleyError!
25             Aviso(Titulo: "Red", Mensaje: "Error de conexión")
26         }
27     )
28     queue.add(stringReq)
29 }
```

Respuestas:

Al dar clic o tap a uno de los comentarios, se despliega un modal con sus respuestas.

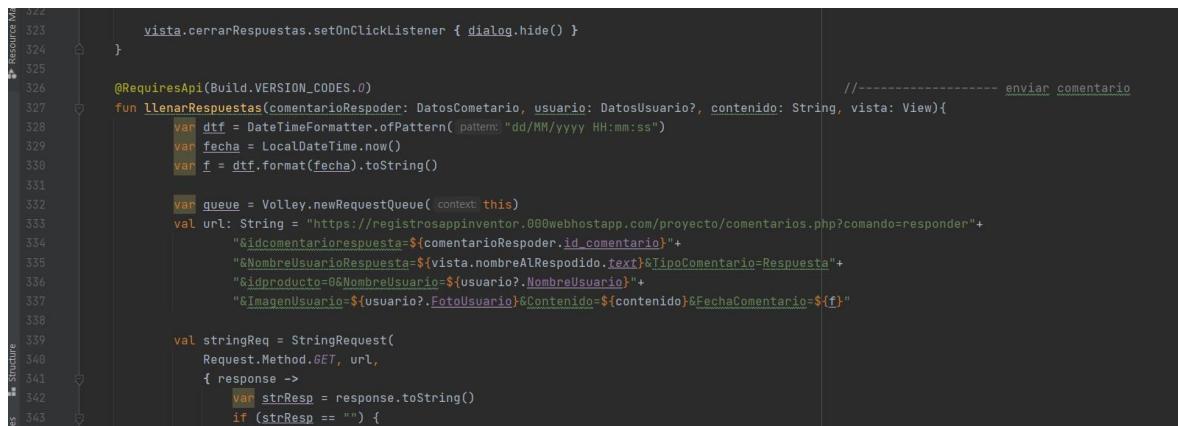


Listar respuestas:



```
378     coment.id_comentario = datos.getString( name: "idcomentario").toString()
379     coment.id_comentario_resposta = datos.getString( name: "idcomentarioresposta").toString()
380     coment.id_producto = datos.getString( name: "idproducto").toString()
381     coment.Tipo_Comentario = datos.getString( name: "TipoComentario").toString()
382     coment.Nombre_Usuario= datos.getString( name: "NombreUsuario").toString()
383     coment.Nombre_Usuario_Respuesta = datos.getString( name: "NombreUsuarioRespuesta").toString()
384     coment.Imagen_Usuario = datos.getString( name: "ImagenUsuario").toString()
385     coment.Contenido = datos.getString( name: "Contenido").toString()
386     coment.Fecha_Comentario = datos.getString( name: "FechaComentario").toString()
387     comentarioProducto.add(coment)
388     indice++
389 }
390
391 //-----Llenado del adaptador y recycler
392
393 vista.recyclerRespuestas.layoutManager = LinearLayoutManager( context: this)
394 val miAdaptador = AdaptadorComentarios(comentarioProducto)
395 vista.recyclerRespuestas.adapter = miAdaptador
396
397 //-----Asignación del clic listener
398 miAdaptador.setOnItemClickListener(object :
399     AdapterView.OnItemClickListener {
400         @RequiresApi(Build.VERSION_CODES.O)
401         override fun onItemClick(position: Int) {
402             vista.nombreAlResedido.text = comentarioProducto.get(position).Nombre_Usuario
403         }
404     })
405
406 }, or
407 { it: VolleyError
408     Aviso( Titulo: "Red", Mensaje: "No se pudo conectar a la base")
409 }
410 }
```

Responder:



```
322     vista.cerrarRespuestas.setOnClickListener { dialog.hide() }
323 }
324
325 @RequiresApi(Build.VERSION_CODES.O)
326 fun llenarRespuestas(comentarioResponder: DatosComentario, usuario: DatosUsuario?, contenido: String, vista: View){
327     var dtf = DateTimeFormatter.ofPattern( pattern: "dd/MM/yyyy HH:mm:ss")
328     var fecha = LocalDateTime.now()
329     var f = dtf.format(fecha).toString()
330
331     var queue = Volley.newRequestQueue( context: this)
332     val url: String = "https://registroappinventor.000webhostapp.com/proyecto/comentarios.php?comando=respoder"+
333         "&idcomentarioresposta=${comentarioResponder.id_comentario}" +
334         "&NombreUsuarioResposta=${vista.nombreAlResedido.text}&TipoComentario=Respuesta" +
335         "&idproducto=&NombreUsuario=${usuario?.NombreUsuario}"+
336         "&ImagenUsuario=${usuario?.FotoUsuario}&Contenido=${contenido}&FechaComentario=${f}"
337
338     val stringReq = StringRequest(
339         Request.Method.GET, url,
340         { response ->
341             var strResp = response.toString()
342             if (strResp == "") {
```

Detalles del producto

A screenshot of a mobile application interface. At the top, there's a header bar with the time (2:16), battery level (100%), and signal strength. Below the header is a back button labeled "Volver" and a shopping cart icon. The main content area features a large image of a sandwich on a wooden board. Below the image, the word "Torta" is displayed in bold. Underneath "Torta", it says "Venta de un producto". The brand name "Umaru Tacos" is listed under "Comida". A five-star rating icon is shown. To the right of the rating, there are two boxes: "Precio: \$ 35" and "Disponibles: 26". Below these boxes is another "Precio de envío: \$ 0" box. At the bottom of the screen are standard mobile navigation icons: back, forward, search, and others.

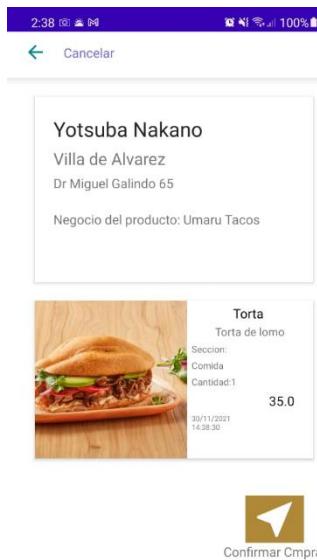
Comentarios.

A screenshot of a mobile application interface showing a comment section. At the top, there's a header bar with the time (2:16), battery level (100%), and signal strength. Below the header is a back button labeled "Volver" and a shopping cart icon. The main content area shows a comment from "Yotsuba Nakano" dated 30/11/2021 at 14:16:10. The comment reads: "Buenas tardes, aparte de la torta, contiene otros aditamentos ? como refresco etc. Xd". Below this comment is another from "Yotsuba Nakano" dated 30/11/2021 at 11:13:49 with the message "uWu". A third comment from "Fujiwara chan" dated 29/11/2021 at 14:45:31 is partially visible. At the bottom, there's a text input field labeled "Comentar..." with a blue send button, and below it are standard mobile navigation icons.

Respuestas:

A screenshot of a mobile application interface showing a direct message between two users. The recipient is "Yotsuba Nakano". The message from "Umaru chan" dated 30/11/2021 at 14:17:47 says: "No, solo es la torta, más tarde publicaré combos, donde ya serán paquetes con múltiples productos, buen día". The response from "Yotsuba Nakano" dated 30/11/2021 at 14:19:08 says: "Ta bien". At the bottom, there's a text input field labeled "Comentar..." with a blue send button, and below it are standard mobile navigation icons.

Confirmar compra



Carrito de compras:

Listar elementos del carrito de compras:

```
CarritoCompras.fkt x
81     alert.setTitle("Aviso!")
82     alert.show()
83 }
84 }
85 fun listarElementosCarrito(usuario: DatosUsuario? ) {
86     elementosCarrito.clear()
87     val queue = Volley.newRequestQueue(activity)
88     val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/carrito_compras.php?comando=ElementosCarrito"+
89         "&correo=${usuario?.CorreoUsuario}"
90
91     val respuestaString = StringRequest(
92         Request.Method.GET, url, { response ->
93             //-----Obtención de un array json con los datos consultados
94             Log.i( tag: "response", response.toString())
95
96             var respuesta = response.toString()
97             val jsonObj: JSONObject = JSONObject(respuesta)
98             val jsonArray: JSONArray = jsonObj.getJSONArray( name: "miCarrito")
99
100            //----- condición de cuando no existan productos
101            if (jsonArray.length() == 0) {
102                elementosCarrito.clear()
103                listaCarrito.layoutManager = LinearLayoutManager(activity)
104                var miAdapter = AdaptadorCarritoCompras(elementosCarrito)
105                listaCarrito.adapter = miAdapter
106                avisoCantidadElementos.setText("Número de elementos: "+ elementosCarrito.size)
107                VaciarCarrito.visibility = View.GONE
108                Aviso( Titulo: "Aviso", Mensaje: "no se encontraron elementos en su carrito de compras")
109            } else {
110
111                var indice = 0
112                //-----Almacenamiento del array en variable, y busqueda en atributos del objeto
113                for (i in 0 until jsonArray.length()) {
```

Eventos: convertir la pantalla de los detalles del producto en un modal y mostrarla:

```
@RequiresApi(Build.VERSION_CODES.O)
fun eventos (elemento: DatosCarrito){
    var modal = AlertDialog.Builder(requireActivity())
    modal.setCancelable(true)
    var vista = LayoutInflater.inflate(R.layout.activity_producto_seleccionado_cliente, root null)

    vista.barreraSuperiorModal.visibility = View.GONE

    Picasso.get().load(elemento.Imagen_Producto).into(vista.imagenProductoConsultado)
    vista.consultaNombreProducto.text = elemento.Nombre_Producto
    vista.consultaTipoOferta.text = elemento.Tipo_Oferta
    vista.consultaNombreNegocio.text = elemento.Nombre_Negocio
    vista.consultaSeccion.text = elemento.Seccion_Producto
    vista.consultaCPrecio.text = "$"+elemento.el_Precio
    vista.consultaCPrecioEnvio.text ="Precio de envío: "+ elemento.Precio_Envio
    vista.consultaExistencias.text = elemento.Existencia_Producto
    vista.CantidadAVender.setText(elemento.Cantidad_Solicitada)

    //----- definir las estrellas
    llamarVotacion(elemento.id_producto.toInt(), vista)

    //definir el estado del boton de los comentarios
    vista.desplegarComentarios.setBackgroundResource(R.drawable.icono_comentario)
    var visibilidadComentarios = false
    vista.cajaLosComentarios.visibility = View.GONE

    modal.setView(vista)
    var dialog = modal.create()
    dialog.show()

    var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil") //----importacion de los datos usuario
```

Edición de cantidades solicitadas:

```
//----- definir las estrellas
llamarVotacion(elemento.id_producto.toInt(), vista)

//definir el estado del boton de los comentarios
vista.desplegarComentarios.setBackgroundResource(R.drawable.icono_comentario)
var visibilidadComentarios = false
vista.cajaLosComentarios.visibility = View.GONE

modal.setView(vista)
var dialog = modal.create()
dialog.show()

var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil") //----importacion de los datos usuario

//----- incremento de la cantidad
var cantidadVenta = vista.CantidadAVender.text.toString().toInt()
var existencias = elemento.Existencia_Producto.toInt()
var precioTotal = elemento.el_Precio.toFloat()
var precioEnvio = elemento.Precio_Envio.toFloat()
var PrecioIndi = (precioTotal - precioEnvio)/cantidadVenta
var pF = (PrecioIndi * cantidadVenta).toFloat()
vista.incremento.setOnClickListener { it: View!
    if(cantidadVenta < existencias){
        cantidadVenta++
        pF = (PrecioIndi * cantidadVenta).toFloat()
        vista.CantidadAVender.setText(cantidadVenta.toString())
        vista.consultaCPrecio.setText("$"+pF)
    }
}
```

Eliminación de un producto:

```
fun carritoEdicionBorrado(elemento: DatosCarrito, comando:String) {
    val queue = Volley.newRequestQueue(activity)
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/carrito_compras.php?comando=${comando}" +
        "&correo=${elemento.correo_Cliente}&idproductocarrito=${elemento.idproductocarrito}&CantidadSolicitada=${elemento.Cantidad_Solicitada}" +
        "&precio=${elemento.el_Precio}&nombre=${elemento.Nombre_Producto}"

    val stringReq = StringRequest(
        Request.Method.GET, url,
        { response ->
            var strResp = response.toString()
            if (strResp == "") {
                Aviso( Titulo: "Producto", Mensaje: "No se pudo editar!")
            } else { //mensaje de respuesta al registrar
                //filtro para no mostrar mensaje cuando solo se cambie el numero
                if(strResp != "editado"){
                    val dialogBuilder = AlertDialog.Builder(requireActivity())
                    dialogBuilder.setMessage(strResp)
                    .setCancelable(false)
                    .setPositiveButton( text: "OK", DialogInterface.OnClickListener { dialog, id ->
                        dialog.cancel()
                    })
                }

                val alert = dialogBuilder.create()
                alert.setTitle("Edición de Productos")
                alert.show()
            }
        }
    )
    var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil") //-----importacion de los datos usuario
}
```

Vaciar carrito:

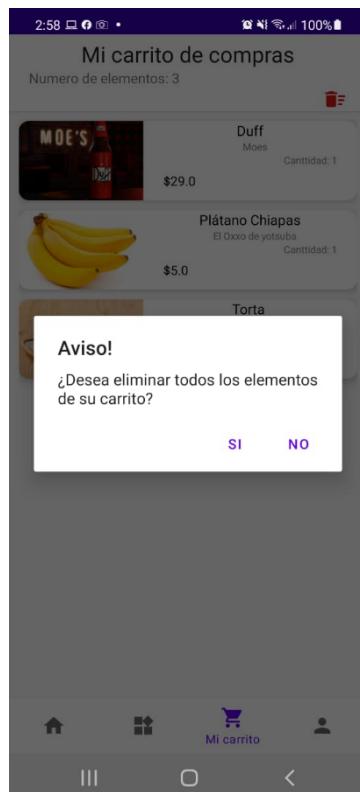
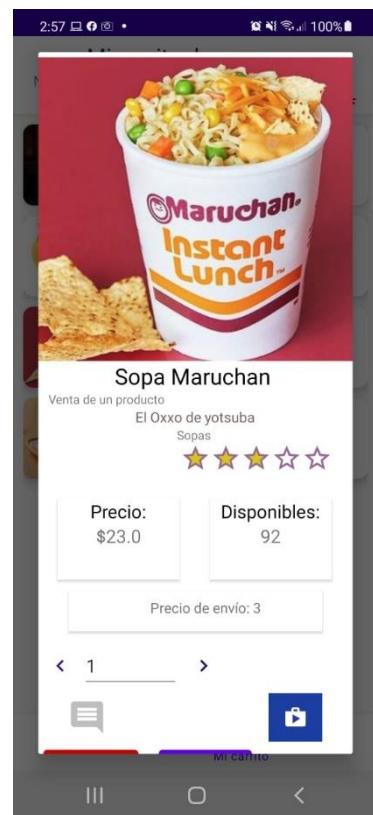
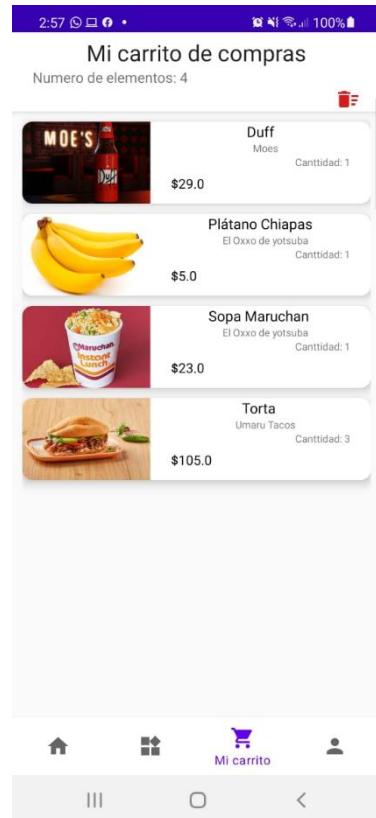
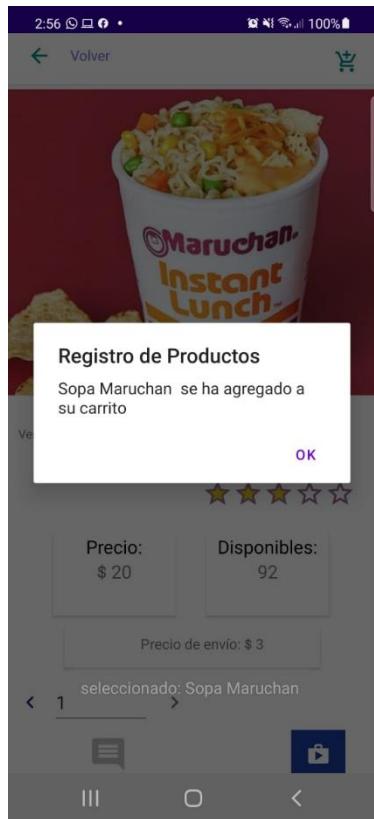
```
fun VaciarElCarro(usuario: DatosUsuario?){
    val queue = Volley.newRequestQueue(activity)
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/carrito_compras.php?comando=vaciarCarrito" +
        "&correo=${usuario?.CorreoUsuario}"

    val stringReq = StringRequest(
        Request.Method.GET, url,
        { response ->
            var strResp = response.toString()
            if (strResp == "") {
                Aviso( Titulo: "Producto", Mensaje: "No se pudo editar!")
            } else { //mensaje de respuesta al registrar

                val dialogBuilder = AlertDialog.Builder(requireActivity())
                dialogBuilder.setMessage(strResp)
                .setCancelable(false)
                .setPositiveButton( text: "OK", DialogInterface.OnClickListener { dialog, id ->
                    dialog.cancel()
                })

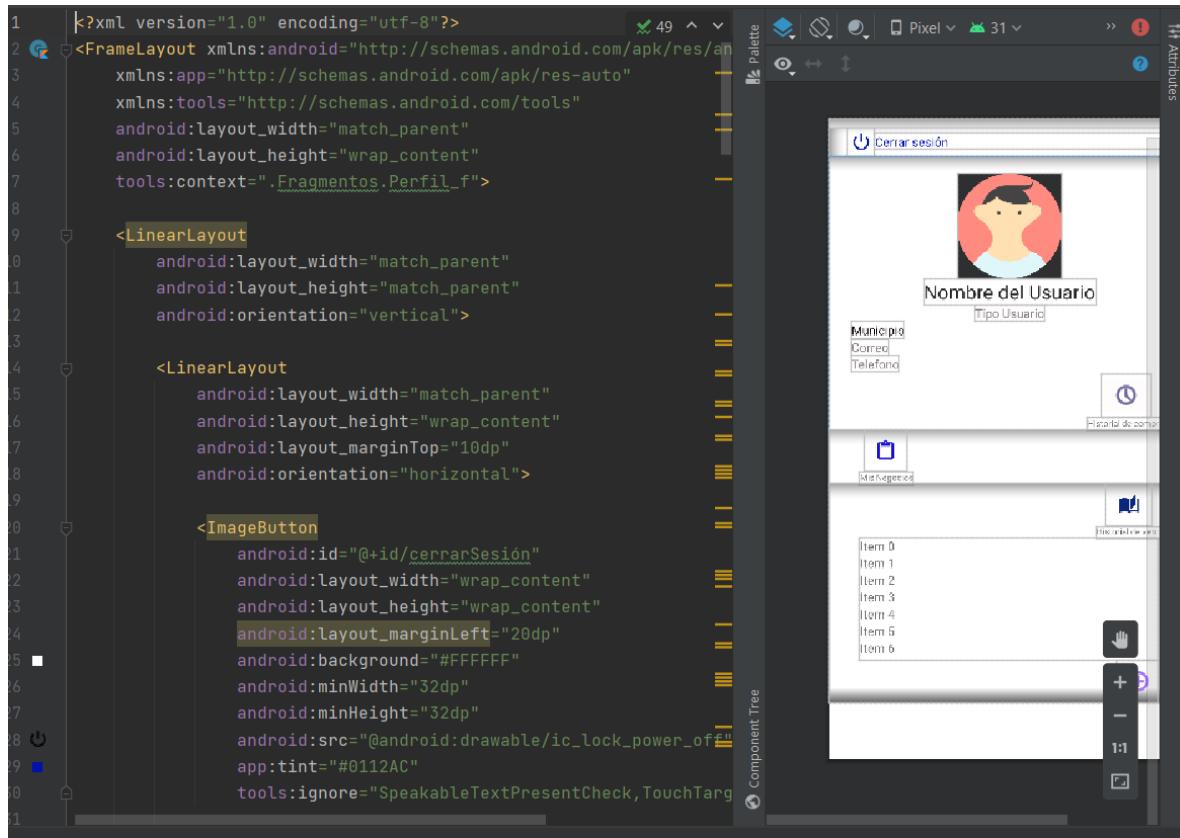
                val alert = dialogBuilder.create()
                alert.setTitle("Edición de Productos")
                alert.show()

            }
        }
    )
    var usuario = arguments?.getParcelable<DatosUsuario>( key: "infoPerfil") //-----importacion de los datos usuario
    listarElementosCarrito(usuario)
}
```



Perfil

Perfil del usuario:



The screenshot shows the Android Studio interface. On the left is the XML code for a fragment named 'Perfil_f'. The code defines a FrameLayout containing a LinearLayout with a vertical orientation. Inside this is another LinearLayout with horizontal orientation, containing an ImageButton with a lock icon and some descriptive text. On the right is the 'Design' tab of the layout editor, showing a preview of the user profile screen. It features a circular profile picture, a name field labeled 'Nombre del Usuario' with a placeholder 'Tipo Usuario', and several input fields for 'Municipio', 'Correo', and 'Telefono'. Below these are sections for 'Historial de compras' and 'Historial de ventas', each with a list of items. At the bottom, there's a navigation bar with icons for Home, Store, Cart, and Profile.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context=".Fragmentos.Perfil_f">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

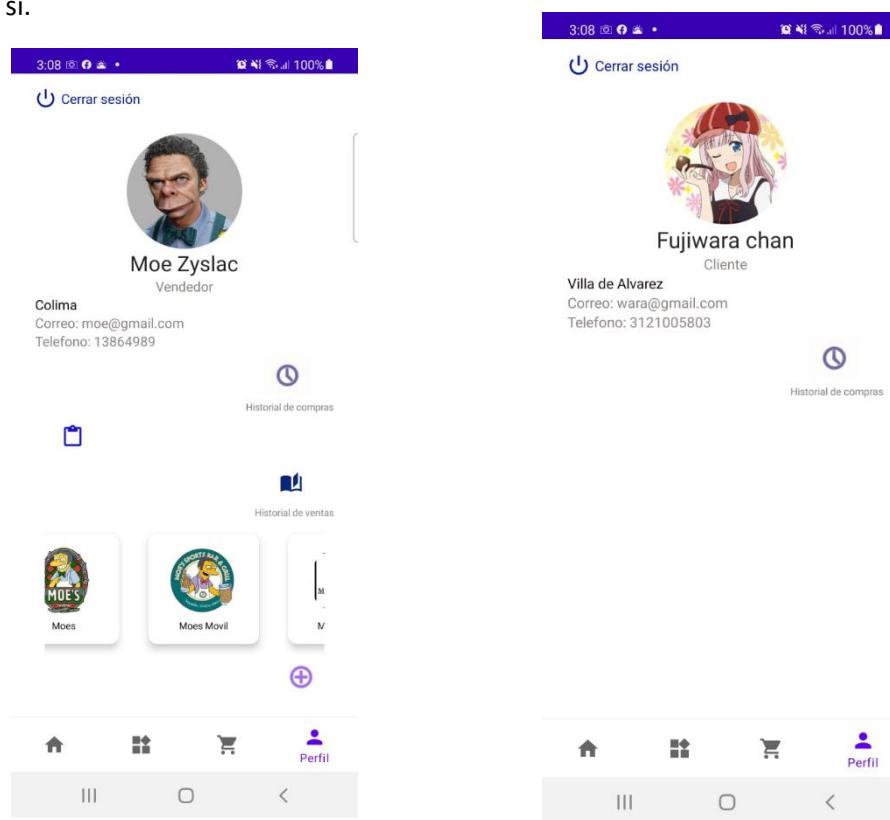
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:orientation="horizontal">

            <ImageButton
                android:id="@+id/cerrarSesion"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="20dp"
                android:background="#FFFFFF"
                android:minWidth="32dp"
                android:minHeight="32dp"
                android:src="@android:drawable/ic_lock_power_off"
                app:tint="#0112AC"
                tools:ignore="SpeakableTextPresentCheck,TouchTargetSizeWarning" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Cerrar sesión" />
        
    

```

Definición de los tipos de usuario; en el cliente no aparecen los negocios del usuario, en el vendedor, si.



```
55
56    //----- Asignación de valores
57    if(usuario?.FotoUsuario?.toString().length > 10){Picasso.get().load(usuario?.FotoUsuario).into(miImagenUsuario)}
58    else{Picasso.get().load( path: "https://w7.pngwing.com/pngs/831/88/png-transparent-user-profile-computer-icons-user-interface-mystique-mysterious-user-profile-transparent-background.png")}
59
60    miNombreUsuario.setText(usuario?.NombreUsuario)
61    miTipoDeUsuario.setText(usuario?.TipoDeUsuario)
62    miMunicipio.setText(usuario?.MunicipioUsuario)
63    miCorreο.setText("Correo: "+usuario?.CorreoUsuario)
64    miTelefono.setText("Teléfono: "+usuario?.TelefonoUsuario)
65
66
67    //----- Desaparecer o Aparecer el botón de administrar negocios
68
69    if(usuario?.TipoDeUsuario == "Vendedor"){accesoNegocios.visibility = View.VISIBLE}
70    else{accesoNegocios.visibility = View.GONE}
71
72
73    //----- observar y listar negocios
74    var esVisible = false
75    misNegociosMiPerfil.visibility = View.GONE
76
77    vistaNegocios.setOnClickListener { it:View|
78        esVisible = !esVisible
79
80        if(esVisible){
81            textView4.visibility = View.GONE
82            misNegociosMiPerfil.visibility = View.VISIBLE
83            listarMisNegocios(usuario)
84        }else{
85            misNegociosMiPerfil.visibility = View.GONE
86            textView4.visibility = View.VISIBLE
87        }
88    }
89
90
```

Cliente:

Historial de compras:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_historial_compras)
    var bar = supportActionBar
    bar?.hide()
    //-----obtener informacion
    var bunle = intent.extras
    var comprador = bunle?.getParcelable<DatosUsuario>( key: "histrialVentas" )

    generarHistorial(comprador)
    cerrarHistorialCompras.setOnClickListener { finish() }
}

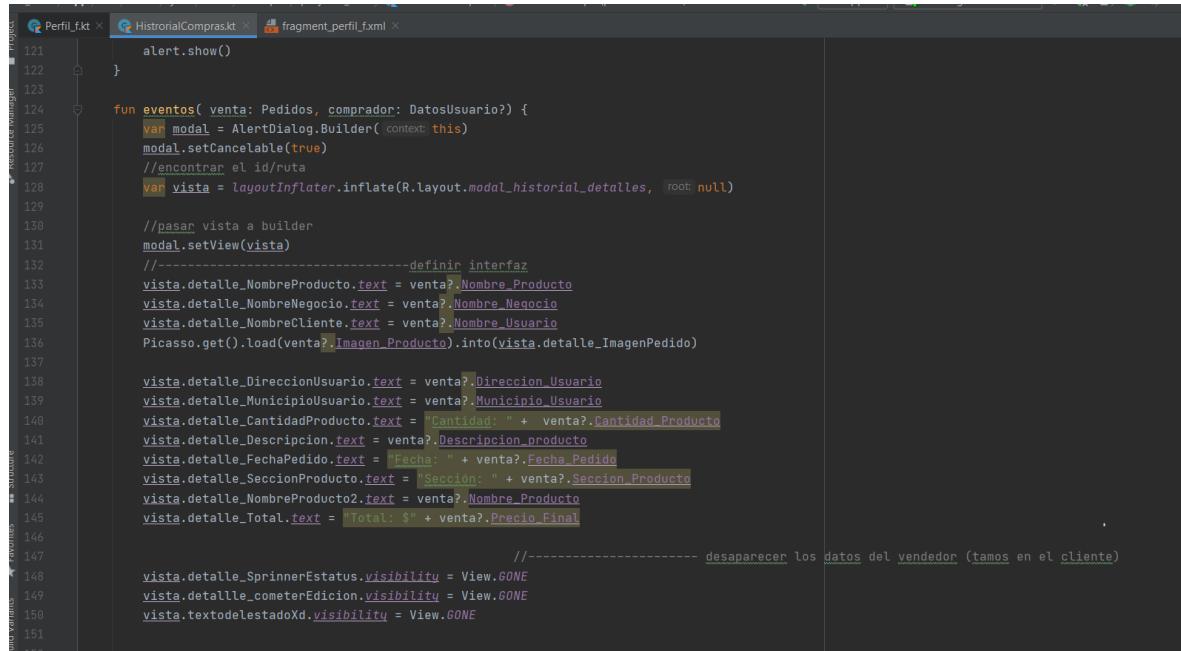
fun generarHistorial(usuario: DatosUsuario?){
    val queue = Volley.newRequestQueue( context: this )
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/pedido.php?comando=listarCliente"+
        "&idusuario=${usuario?.idusuario}"

    val respuestaString = StringRequest(
        Request.Method.GET, url, { response ->
            //-----Obtención de un array json con los datos consultados
            Log.i( tag: "response", response.toString() )

            var respuesta = response.toString()
            val jsonObj: JSONObject = JSONObject(respuesta)
            val jsonArray: JSONArray = jsonObj.getJSONArray( name: "pedidoscliente" )
            if ( jsonArray.length() == 0 ) {
                Aviso( Titulo: "Aviso", Mensaje: "No cuenta con compras registradas" )
            } else {
                var misComprasListadas = ArrayList<Pedidos>()
                var indice = 0

                //-----Almacenamiento del array en variable, y busqueda en atributos del objeto
            }
        }
    )
}
```

Modal de detalles pedido:



The screenshot shows the Android Studio interface with the code editor open. The file is named `fragment_perfil_fxml`. The code is in Java and defines a modal dialog for viewing order details. It uses the `AlertDialog.Builder` class to create the modal, sets it as cancelable, and inflates a layout from `R.layout.modal_historial_detalles`. The inflated view is then used to set various text fields with data from the `venta` object. The code includes comments explaining the purpose of certain sections, such as setting the view and defining the interface. At the bottom, there is a note about hiding vendor information.

```
121     alert.show()
122 }
123
124 fun eventos( venta: Pedidos, comprador: DatosUsuario?) {
125     var modal = AlertDialog.Builder( context: this)
126     modal.setCancelable(true)
127     //encontrar el id/ruta
128     var vista = layoutInflater.inflate(R.layout.modal_historial_detalles, root: null)
129
130     //pasar vista a builder
131     modal.setView(vista)
132     //-----definir interfaz
133     vista.detalle_NombreProducto.text = venta?.Nombre_Producto
134     vista.detalle_NombreNegocio.text = venta?.Nombre_Negocio
135     vista.detalle_NombreCliente.text = venta?.Nombre_Usuario
136     Picasso.get().load(venta?.Imagen_Producto).into(vista.detalle_ImagenPedido)
137
138     vista.detalle_DireccionUsuario.text = venta?.Direccion_Usuario
139     vista.detalle_MunicipioUsuario.text = venta?.Municipio_Usuario
140     vista.detalle_CantidadProducto.text = "Cantidad: " + venta?.Cantidad_Producto
141     vista.detalle_Descripcion.text = venta?.Descripcion_producto
142     vista.detalle_FechaPedido.text = "Fecha: " + venta?.Fecha_Pedido
143     vista.detalle_SectionProducto.text = "Sección: " + venta?.Section_Producto
144     vista.detalle_NombreProducto2.text = venta?.Nombre_Producto
145     vista.detalle_Total.text = "Total: $" + venta?.Precio_Final
146
147     //----- desaparecer los datos del vendedor (tamos en el cliente)
148     vista.detalle_SprinnerEstatus.visibility = View.GONE
149     vista.detalle_cometerEdicion.visibility = View.GONE
150     vista.textodelestadoXd.visibility = View.GONE
151 }
```

Cancelar pedido:

```
var dialog = modal.create()
dialog.show()

vista.cancelar_deatalles.setOnClickListener { it: View!
    dialog.hide()
}

//----- cancelar compra
vista.cancelar_compra.setOnClickListener { it: View!

    val dialogBuilder = AlertDialog.Builder( context: this)
    dialogBuilder.setMessage("¿Desea confirmar la cancelación del pedido?")
        .setCancelable(false)
        .setPositiveButton( text: "si", DialogInterface.OnClickListener {
            modalConfirmacion, id -> modalConfirmacion.cancel()
            dialog.hide()
            cancelarCompra(venta, comprador)
        }).setNegativeButton( text: "No", DialogInterface.OnClickListener{
            modalConfirmacion, id -> modalConfirmacion.cancel()
        })
    val alert = dialogBuilder.create()
    alert.setTitle("Aviso")
    alert.show()
}

}

}

queue.add(respuestaString)

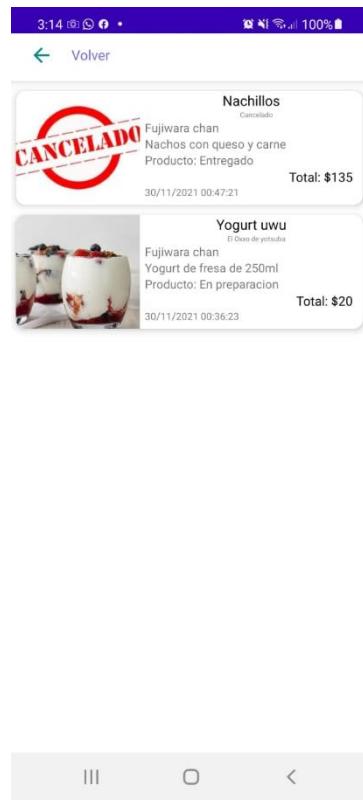
}

fun editarVotacion(pedido: Pedidos, comprador: DatosUsuario?, Calificacion: Int){
    val queue = Volley.newRequestQueue( context: this)

    var url: String = "https://registrosappinventor.000webhostapp.com/proyecto/votaciones.php?comando=EditarPuntuacion"+
        "&idnegocio=${pedido?.id_negocio}&idproducto=${pedido.id_producto}&correo=${comprador?.CorreoUsuario}&Calificacion=${Calificacion}"

    val stringReq = StringRequest(
        Request.Method.GET, url,
        { response ->
            var strResp = response.toString()
            if (strResp == "") {
                //Aviso("Producto","No se pudo editar!")
            } else { //mensaje de respuesta al registrar
                //Aviso("aviso", strResp)
            }
        },
        { it: VolleyError?
            Aviso( Titulo: "Red", Mensaje: "Error de conexión")
        }
    )
    queue.add(stringReq)
}
```

Calificación, votos por estrellas:



Historial de ventas (este apartado solo está disponible para el vendedor):

```
182
183     when(elEstado){
184         "En preparación" -> indice = 0
185         "En camino" -> indice = 1
186         "Entregado" -> indice = 2
187         "Realizándose" -> indice = 0
188         "Terminado" -> indice = 1
189     }
190     //-----
191     var adaptador = ArrayAdapter<String>( context: this, android.R.layout.simple_gallery_item, estadosPedido)
192     vista.detalle_SpinnerEstatus.adapter = adaptador
193     vista.detalle_SpinnerEstatus.setSelection(indice)
194
195     var dialog = modal.create()
196     dialog.show()
197
198     //-----
199     vista.textViewVotacion.visibility = View.GONE
200     vista.puntuacion.visibility = View.GONE
201
202     vista.cancelar_deatalles.setOnClickListener { dialog.hide() }
203
204     if(detalie.Nombre_Negocio == "Cancelado"){
205         vista.textodelestadoXd.visibility = View.GONE
206         vista.detalle_SpinnerEstatus.visibility = View.GONE
207         vista.detalle_cometerEdicion.visibility = View.GONE
208     }
209     //-----
210     //----- editar y refrescar de paso
211     vista.detalle_cometerEdicion.setOnClickListener{ it:View!
212         editarEstatus(detalie, vista.detalle_SpinnerEstatus.selectedItem.toString())
213
214         var bunle = intent.extras
215         var vendedor = bunle?.getParcelable<DatosUsuario>( key: "histrialVentas")
```

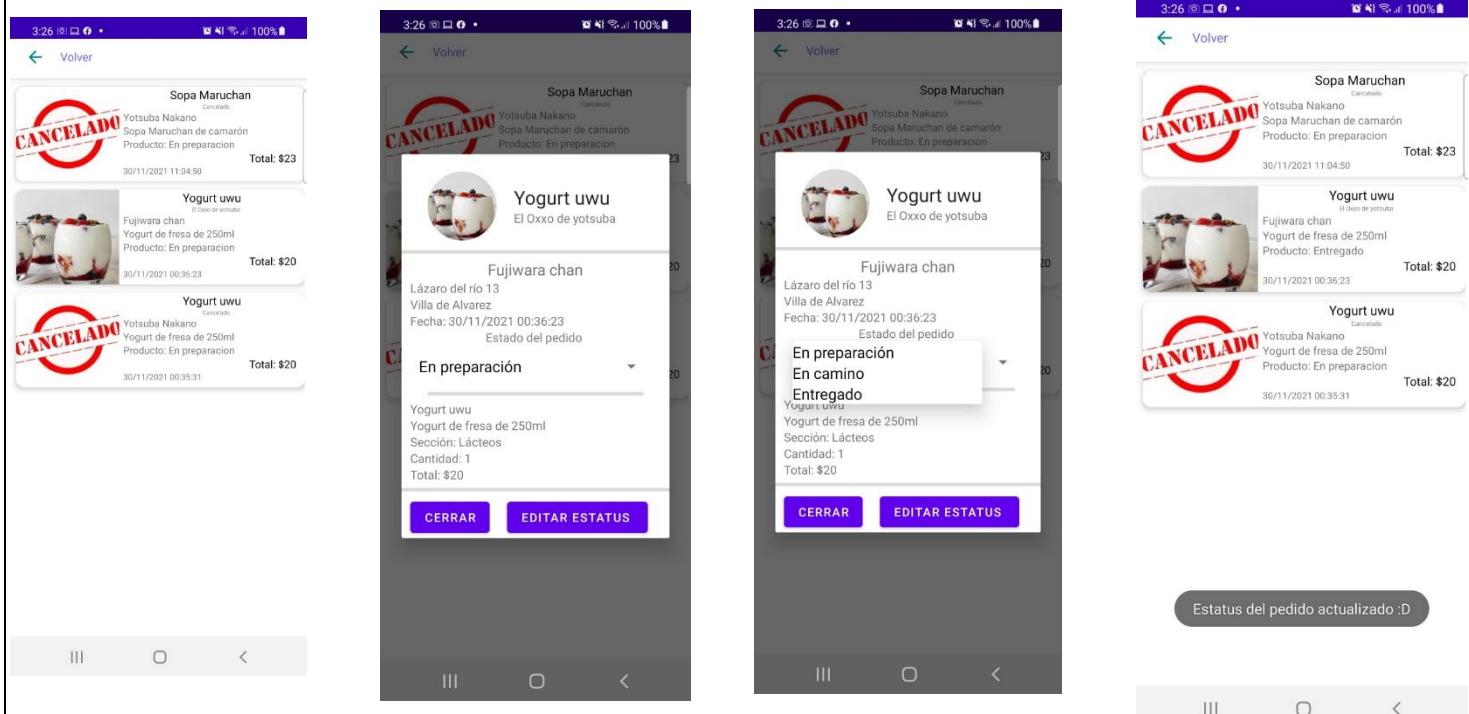
Modal detalles pedido vendedor: (el mismo que detalles pedido cliente, solo oculta las votaciones y hago visible el cambio de estado y el de cometer edición).

```
    }
}

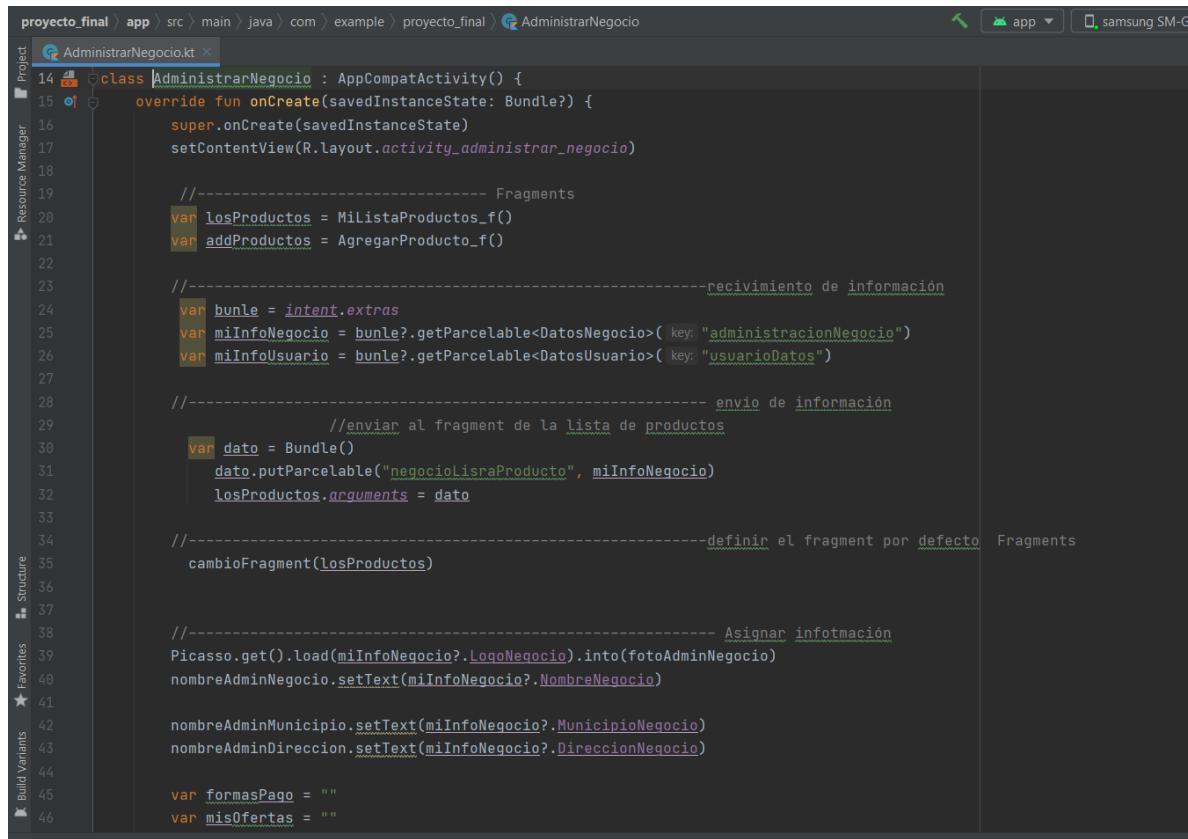
fun editarEstatus(idPedidos: Pedidos?, estatus:String)
{
    val queue = Volley.newRequestQueue( context: this)
    var url: String = "https://registrosappinventor.000webhostapp.com/proyecto/pedido.php?comando=EditarEstatus"+
        "&idpedido=${idPedidos?.id_pedido}&estatus=${estatus}"

    val stringReq = StringRequest(
        Request.Method.GET, url,
        { response ->
            var strResp = response.toString()
            if (strResp == "") {
                Aviso( Titulo: "Producto", Mensaje: "No se pudo editar!")
            } else { //mensaje de respuesta al registrar
                Toast.makeText( context: this, text: "$strResp", Toast.LENGTH_SHORT).show()
            }
        },
        { it: VolleyError!
            Aviso( Titulo: "Red", Mensaje: "Error de conexión")
        }
    )
    queue.add(stringReq)
}
```

Se pude apreciar como la cancelación afecta la lista de historial de ventas del vendedor:



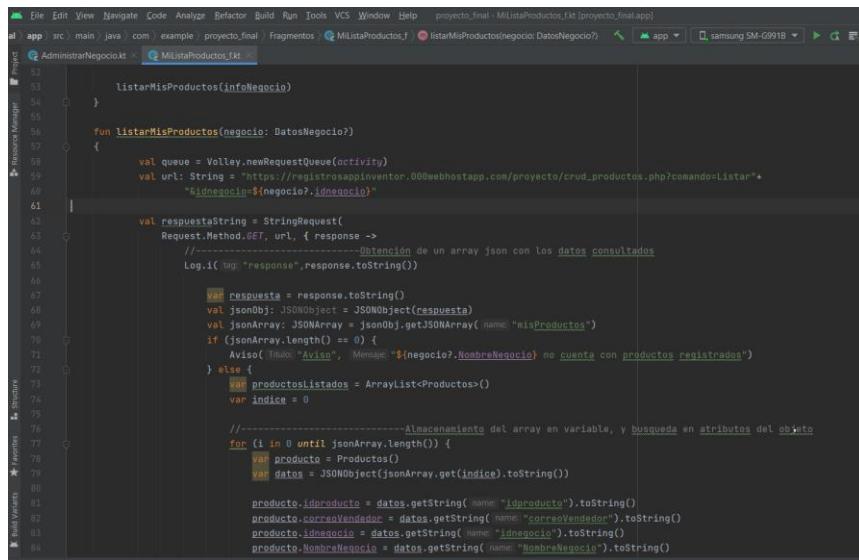
Negocios del cliente: al dar tap al negocio, aparecen los detalles del negocio en cuestión.



```
project_final > app > src > main > java > com > example > proyecto_final > AdministrarNegocio
```

```
14 package com.example.proyecto_final;
15 import android.os.Bundle;
16 import androidx.appcompat.app.AppCompatActivity;
17 import android.view.View;
18
19 public class AdministrarNegocio extends AppCompatActivity {
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_administrar_negocio);
24
25         //----- Fragments
26         var losProductos = MilistaProductos_f();
27         var addProductos = AgregarProducto_f();
28
29         //-----recivimiento de información
30         var bunle = intent.getExtras();
31         var miInfoNegocio = bunle.getParcelable<DatosNegocio>( key: "administracionNegocio" );
32         var miInfoUsuario = bunle.getParcelable<DatosUsuario>( key: "usuarioDatos" );
33
34         //----- envio de información
35         //enviar al fragment de la lista de productos
36         var dato = Bundle();
37         dato.putParcelable("negocioListaProducto", miInfoNegocio);
38         losProductos.arguments = dato;
39
40         //-----definir el fragment por defecto
41         cambioFragment(losProductos);
42
43         //----- Asignar infotmación
44         Picasso.get().load(miInfoNegocio.LogoNegocio).into(fotoAdminNegocio);
45         nombreAdminNegocio.setText(miInfoNegocio.NombreNegocio);
46
47         nombreAdminMunicipio.setText(miInfoNegocio.MunicipioNegocio);
48         nombreAdminDireccion.setText(miInfoNegocio.DireccionNegocio);
49
50         var formasPago = "";
51         var misOfertas = "";
```

Fragment de la lista de los productos:



```
project_final > app > src > main > java > com > example > proyecto_final > Fragmentos > MilistaProductos_f > listarMisProductos(negocio: DatosNegocio)
```

```
13 package com.example.proyecto_final.Fragmentos;
14 import android.os.Bundle;
15 import android.util.Log;
16 import android.view.LayoutInflater;
17 import android.view.View;
18 import android.view.ViewGroup;
19 import androidx.fragment.app.Fragment;
20 import androidx.recyclerview.widget.LinearLayoutManager;
21 import androidx.recyclerview.widget.RecyclerView;
22
23 import com.example.proyecto_final.R;
24
25 public class MilistaProductos_f extends Fragment {
26     private RecyclerView recyclerView;
27     private MilistaProductosAdapter adapter;
28
29     @Override
30     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
31         View view = inflater.inflate(R.layout.fragment_milista_productos, container, false);
32
33         recyclerView = view.findViewById(R.id.recyclerView);
34         recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
35
36         return view;
37     }
38
39     public void listarMisProductos(DatosNegocio negocio) {
40         val queue = Volley.newRequestQueue(activity);
41         val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/crud_productos.php?comando=listar"+
42             "&idnegocio=$negocio.idnegocio";
43
44         val respuestaString = StringRequest(
45             Request.Method.GET, url, { response ->
46                 //Obtención de un array json con los datos consultados
47                 Log.i("response", response.toString());
48
49                 val jsonObj: JSONObject = JSONObject(response);
50                 val jsonArray: JSONArray = jsonObj.getJSONArray("misProductos");
51                 if (jsonArray.length() == 0) {
52                     Aviso("Aviso", "El negocio ${negocio.NombreNegocio} no cuenta con productos registrados");
53                 } else {
54                     var productosListados = ArrayList<Productos>();
55                     var indice = 0;
56
57                     //Almacenamiento del array en variable, y búsqueda en atributos del objeto
58                     for (i in 0 until jsonArray.length()) {
59                         var producto = Productos();
60                         var datos = JSONObject(jsonArray.getJSONObject(indice));
61
62                         producto.idproducto = datos.getString("idproducto");
63                         producto.correoVendedor = datos.getString("correoVendedor");
64                         producto.idnegocio = datos.getString("idnegocio");
65                         producto.NombreNegocio = datos.getString("NombreNegocio");
```

editar, borrar y crear productos:

```
final | app > src > main > java > com > example > proyecto_final > Fragmentos > EditarBorrarProductos_f.kt
```

```
    } else { //mensaje de respuesta al registrar

        val dialogBuilder = AlertDialog.Builder(requireActivity())
        dialogBuilder.setMessage(strResp)
            .setCancelable(false)
            .setPositiveButton("OK", DialogInterface.OnClickListener { dialog, id ->
                dialog.cancel()
            })

        val alert = dialogBuilder.create()
        alert.setTitle("Edición de Productos")
        alert.show()

    }

}

{
    it: VolleyError
    Aviso( Título: "Rød", Mensaje: "Error de conexión")

}

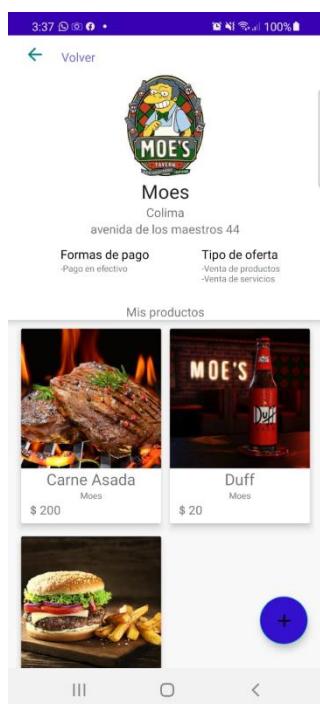
queue.add(stringReq)
}

fun borrarMiProducto(producto: Productos?){
    var queue = Volley.newRequestQueue(activity)
    val url: String = "https://registrosappinventor.000webhostapp.com/proyecto/crud_productos.php?comando=Eliminar"+
        "&idproducto=${producto?.idproducto}"

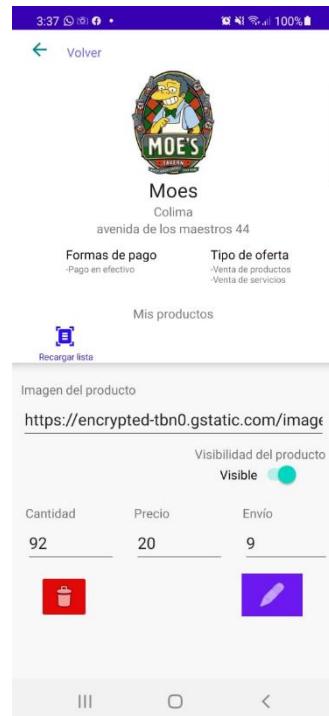
    val stringReq = StringRequest(
        Request.Method.GET, url,
        { response ->
            var strResp = response.toString()
            if (strResp == "") {


```

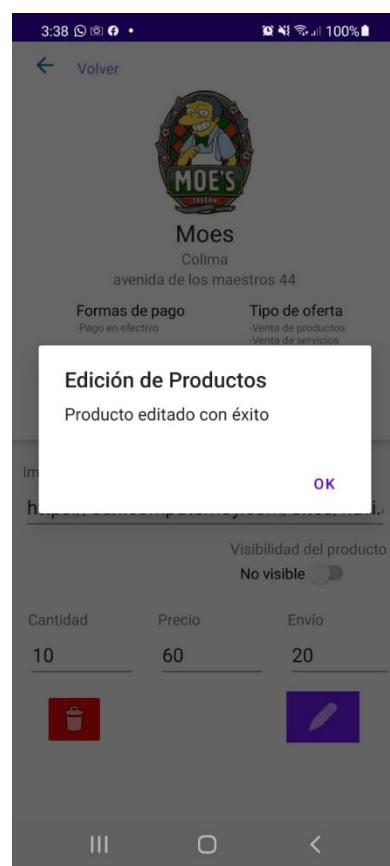
Vista del negocio:

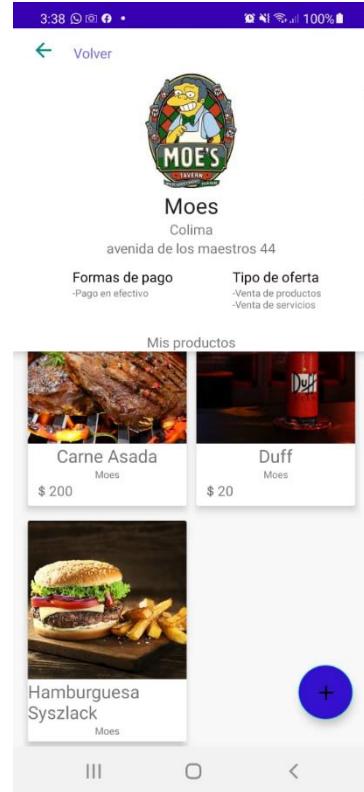


detalles del producto:



edición del producto





APIS :

Login:

```

loginproyecto.php > loginproyecto.php
Proyecto_final > loginproyecto.php
4 $servername = "localhost";
5 $username = "id17488564_proyecto_final";
6 $password = "M%{A|e&$3u^3E|ev";
7 $dbname = "id17488564_proyectofinal";
8
9
10
11 $correo=$_GET["correo"];
12 $contraseña=$_GET["contraseña"];
13
14 $conn = new mysqli($servername, $username, $password, $dbname);
15 // Check connection
16 if ($conn->connect_error)
17 {
18 | die("Conexión fallida: " . $conn->connect_error);
19 }
20
21 $sql = "SELECT * FROM Usuarios where CorreoUsuario='$correo' and ContrasenaUsuario='$contraseña'";
22 $result = $conn->query($sql);
23 $registros=array();
24 $i=0;
25
26 if ($result->num_rows > 0)
27 {
28
29     while($row = $result->fetch_assoc())
30     {
31
32         $registros[$i]=$row;
33         | $i++;
34     }
35     echo '{"usuario":'.json_encode($registros).'}';
36     $conn->close();
37 }
38 else
39 echo '{"records":[]}';
40 ?>

```

Listar negocios del perfil

```
loginproyecto.php listar.php
Proyecto_final > listar.php
1
2
3
4 $servername = "localhost";
5 $username = "id17488564_proyecto_final";
6 $password = "%{A}e$3u^3F|ev";
7 $dbname = "id17488564_proyectofinal";
8
9
10 $comandos=$_GET["comando"];
11 $correo=$_GET["correo"];
12
13 $conn = new mysqli($servername, $username, $password, $dbname);
14 // check connection
15 if ($conn->connect_error)
16 {
17 die("Conexión fallida: " . $conn->connect_error);
18 }
19
20 //----- listar los negocios del perfil
21 if($comando == "NegociosPerfil")
22 {
23 $sql = "SELECT * FROM Negocios where CorreoUsuario='$correo'";
24 $result = $conn->query($sql);
25
26 if ($result->num_rows > 0) {
27 $registros=array();
28 $i=0;
29 while($row = $result->fetch_assoc()) {
30 $registros[$i]=$row;
31 $i++;
32 }
33 echo ('{"misNegocios":'.json_encode($registros).'}');
34 } else {
35 echo "0 resultados";
36 }
37 }
38 $conn->close();
39 ?>
```

Registro del cliente y vendedor:

```
... loginproyecto.php listar.php registrar.php
Proyecto_final > registrar.php
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
if($TipoUsuario == 'Cliente')
{
    $sql = "INSERT INTO Usuarios (NombreUsuario, FotoUsuario, MunicipioUsuario,
                                ColoniaUsuario, DomicilioUsuario, CorreoUsuario,
                                ContrasenaUsuario, TelefonoUsuario,
                                TipoDeUsuario, VerificacionUsuario)
VALUES ('$NombreUsuario', '$FotoUsuario', '$MunicipioUsuario',
        '$ColoniaUsuario', '$DomicilioUsuario', '$CorreoUsuario',
        '$ContrasenaUsuario', '$TelefonoUsuario',
        '$TipoUsuario', '$VerificacionUsuario')";

if( $conn->query($sql)== TRUE)
{
    echo "Nuevo usuario registrado";
}
else {echo "Error: " . $sql . "<br>" . $conn->error;}
}

if($TipoUsuario == "Vendedor")
{
    //variables del negocio
    $NombreNegocio=$_GET["NombreNegocio"];
    $LogoNegocio=$_GET["LogoNegocio"];
    $Efectivo=$_GET["Efectivo"];
    $Tarjeta=$_GET["Tarjeta"];
    $PagoOxxo=$_GET["PagoOxxo"];
    $OfertaProducto=$_GET["OfertaProducto"];
    $OfertaServicio=$_GET["OfertaServicio"];
    $CategoriaNegocio=$_GET["CategoriaNegocio"];
    $MunicipioNegocio=$_GET["MunicipioNegocio"];
    $ColoniaNegocio=$_GET["ColoniaNegocio"];
    $DireccionNegocio=$_GET["DireccionNegocio"];
    //registro del vendedor
    $sql = "INSERT INTO Usuarios (NombreUsuario, FotoUsuario, MunicipioUsuario,
                                ColoniaUsuario, DomicilioUsuario, CorreoUsuario,
                                ContrasenaUsuario, TelefonoUsuario,
                                TipoDeUsuario, VerificacionUsuario)
VALUES ('$NombreUsuario', '$FotoUsuario', '$MunicipioUsuario',
        '$ColoniaUsuario', '$DomicilioUsuario', '$CorreoUsuario',
        '$ContrasenaUsuario', '$TelefonoUsuario',
        '$TipoUsuario', '$VerificacionUsuario')";
```

Registro del pedido:

```
loginproyecto.php    listar.php    registrar.php    pedidos.php X
Proyecto_final > pedidos.php

18 if($comando == "realizarVenta")
19 {
20     $idusuario=$_GET["idusuario"];
21     $correoVendedor=$_GET["correoVendedor"];
22     $NombreUsuario=$_GET["NombreUsuario"];
23     $MunicipioUsuario=$_GET["MunicipioUsuario"];
24     $DireccionUsuario=$_GET["DireccionUsuario"];
25     $idnegocio=$_GET["idnegocio"];
26     $NombreNegocio=$_GET["NombreNegocio"];
27     $idproducto=$_GET["idproducto"];
28     $NombreProducto=$_GET["NombreProducto"];
29     $DescripcionProducto=$_GET["DescripcionProducto"];
30     $SeccionProducto=$_GET["SeccionProducto"];
31     $ImagenProducto=$_GET["ImagenProducto"];
32     $CantidadProducto=$_GET["CantidadProducto"];
33     $PrecioFinal=$_GET["PrecioFinal"];
34     $TipoPedido=$_GET["TipoPedido"];
35     $FechaPedido=$_GET["FechaPedido"];
36
37
38     $sql = "INSERT INTO Pedidos (idusuario, correoVendedor, NombreUsuario, MunicipioUsuario, DireccionUsuario,
39                             idnegocio, NombreNegocio, idproducto, NombreProducto, DescripcionProducto, SeccionProducto,
40                             ImagenProducto, CantidadProducto, PrecioFinal, TipoPedido, FechaPedido)
41
42         VALUES('$idusuario', '$correoVendedor', '$NombreUsuario', '$MunicipioUsuario', '$DireccionUsuario',
43             '$idnegocio', '$NombreNegocio', '$idproducto', '$NombreProducto', '$DescripcionProducto', '$SeccionProducto',
44             '$ImagenProducto', '$CantidadProducto', '$PrecioFinal', '$TipoPedido', '$FechaPedido')";
45
46     $sql2 = "INSERT INTO PedidosCliente (idusuario, NombreUsuario, MunicipioUsuario, DireccionUsuario,
47                                         idnegocio, NombreNegocio, idproducto, NombreProducto, DescripcionProducto, SeccionProducto,
48                                         ImagenProducto, CantidadProducto, PrecioFinal, TipoPedido, FechaPedido)
49
50         VALUES('$idusuario', '$correoVendedor', '$NombreUsuario', '$MunicipioUsuario', '$DireccionUsuario',
51             '$idnegocio', '$NombreNegocio', '$idproducto', '$NombreProducto', '$DescripcionProducto', '$SeccionProducto',
52             '$ImagenProducto', '$CantidadProducto', '$PrecioFinal', '$TipoPedido', '$FechaPedido')";
53
54
55
56     if($conn->query($sql)==true && $conn->query($sql2)==true){echo "Compra realizada con éxito";}
57     else { echo "Error: " . $sql . "<br>" . $conn->error;}
58 }
59
60
61     $conn->close();
62 ?>
```

Puntuaciones

```
... loginproyecto.php listar.php registrar.php pedidos.php puntuaciones.php x

Proyecto_final > puntuaciones.php
final
final

5 $username = "id17488564_proyecto_final";
6 $password = "M%{A|&3\$^3E|eV";
7 $dbname = "id17488564_proyectofinal";
8
9 $comando=$_GET["comando"];
10
11 $conn = new mysqli($servername, $username, $password, $dbname);
12 // Check connection
13 if ($conn->connect_error){
14 die("Conexión fallida: " . $conn->connect_error);
15 }
16
17 if($comando == "votar"){
18     $idnegocio=$_GET["idnegocio"];
19     $idproducto=$_GET["idproducto"];
20     $correoPuntuador=$_GET["correoPuntuador"];
21     $Calificacion=$_GET["Calificacion"];
22
23     $sql = "INSERT INTO Puntuaciones(idnegocio, idproducto
24                                     correoPuntuador, Calificacion)
25                                     Values ('$idnegocio', '$idproducto',
26                                     '$correoPuntuador', '$Calificacion')";
27
28     if($conn->query($sql)===true){echo "Puntuado";}
29     else echo {"error: " . $sql . ' ' . $conn->error.''};
30 }
31
32 if($comando == "EditarPuntuacion")
{
33     $idpuntuacion=$_GET["idpuntuacion"];
34     $idnegocio=$_GET["idnegocio"];
35     $idproducto=$_GET["idproducto"];
36     $correo=$_GET["correo"];
37     $Calificacion=$_GET["Calificacion"];
38
39     $sql = "UPDATE Puntuaciones SET Calificacion='$Calificacion'
40                                     WHERE correoPuntuador='$correo' AND idproducto='$idproducto'";
41
42
43
44
45     if($conn->query($sql)===true){echo "Editado $Calificacion";}
46     else { echo {"error: " . $sql . ' ' . $conn->error.''};}
47 }
48
49 $conn->close();
50 >>
```

Crud de los productos.

```
loginproyecto.php listar.php registrar.php pedidos.php puntuaciones.php crud_productos.php

Proyecto_final > crud_productos.php
51     $i=0;
52     while($row = $result->fetch_assoc()) {
53         $registros[$i]=$row;
54         $i++;
55     }
56     echo '{"misProductos":'.json_encode($registros).'}';
57 } else {
58     echo '{"misProductos":[]}' ;
59 }
60 }
61
62 //----- Editar productos
63 if($comando == "Editar")
64 {
65     $idproducto=$_GET["idproducto"];
66     $NombreProducto=$_GET["NombreProducto"];
67     $SeccionProducto=$_GET["SeccionProducto"];
68     $ExistenciaProducto=$_GET["ExistenciaProducto"];
69     $DescripcionProducto=$_GET["DescripcionProducto"];
70     $TipoOferta=$_GET["TipoOferta"];
71     $ImagenProducto=$_GET["ImagenProducto"];
72     $Precio=$_GET["Precio"];
73     $PrecioEnvio=$_GET["PrecioEnvio"];
74
75     $sql = "UPDATE Productos SET NombreProducto='$NombreProducto',
76                               SeccionProducto='$SeccionProducto',
77                               ExistenciaProducto= '$ExistenciaProducto',
78                               DescripcionProducto= '$DescripcionProducto',
79                               TipoOferta= '$TipoOferta',
80                               ImagenProducto= '$ImagenProducto',
81                               Precio= '$Precio',
82                               PrecioEnvio= '$PrecioEnvio'
83
84                               WHERE idproducto= '$idproducto'";
85
86     if( $conn->query($sql)== TRUE) {echo "Producto editado con éxito";}
87     else {echo "Error: " . $sql . "<br>" . $conn->error;}
88
89 }
90 //----- Eliminación de producto
91 if($comando == "Eliminar")
92 {
93     $idproducto=$_GET["idproducto"];
94 }
```

Comentarios:

```
Proyecto_final > comentarios.php
1  <?php
2  header("Content-type: application/json; charset=utf-8");
3
4  $servername = "localhost";
5  $username = "id17488564_proyecto_final";
6  $password = "M%{A}e&$3u^3E|eV";
7  $dbname = "id17488564_proyectofinal";
8
9  $comando=$_GET["comando"];
10
11 $conn = new mysqli($servername, $username, $password, $dbname);
12 // Check connection
13 if ($conn->connect_error)
14 {
15 | die("Conexión fallida: " . $conn->connect_error);
16 }
17
18 if($comando=="comentar"){
19     $idproducto=$_GET["idproducto"];
20     $TipoComentario=$_GET["TipoComentario"];
21     $NombreUsuario=$_GET["NombreUsuario"];
22     $ImagenUsuario=$_GET["ImagenUsuario"];
23     $Contenido=$_GET["Contenido"];
24     $FechaComentario=$_GET["FechaComentario"];
25
26     $sql = "INSERT INTO ComentarioRespuesta(idproducto, TipoComentario, NombreUsuario,
27                                         ImagenUsuario, Contenido, FechaComentario)
28
29                                         VALUES('$idproducto','$TipoComentario','$NombreUsuario',
30                                         '$ImagenUsuario','$Contenido', '$FechaComentario')";
31
32     if($conn->query($sql)==TRUE){echo "Comentado";}
33     else {echo {"error: " . $sql . " " . $conn->error.""};}
34 }
35 $conn->close();
36 ?>
```

Carrito de compras:

/public_html/proyecto/carrito_compras.php

```
    '
71  |
72  if($comando == "EditarCantidadSolicitada") //----- editar la cantidad
73  {
74      $idproductocarrito=$_GET["idproductocarrito"];
75      $CantidadSolicitada=$_GET["CantidadSolicitada"];
76      $precio=$_GET["precio"];
77
78      $sql = "UPDATE CarritoVentas SET CantidadSolicitada='$CantidadSolicitada',
79          Precio='$precio'
80          WHERE idproductocarrito='$idproductocarrito'";
81
82      if($conn->query($sql) === TRUE) {echo "editado";}
83      else{echo "Error: " . $sql . "<br>" . $conn->error;}
84  }
85
86  |----- Borrar un solo elemento del carrito
87  if($comando == "BorrarElemento")
88  {
89      $idproductocarrito=$_GET["idproductocarrito"];
90      $nombre=$_GET["nombre"];
91
92      $sql = "DELETE FROM CarritoVentas WHERE idproductocarrito='$idproductocarrito'";
93
94      if($conn->query($sql) === TRUE) {echo "'$nombre' eliminado de su carrito";}
95      else{echo "Error: " . $sql . "<br>" . $conn->error;}
96  }
```