



## Título del Proyecto:

Gestión de pedidos

## Autor:

Chisvert Amat,  
Víctor Manuel

## Director:

Carbonell, Vicente

**TESINA PARA LA  
OBTENCIÓN DEL TÍTULO DE:**

**Máster en Desarrollo de  
Aplicaciones sobre Dispositivos  
Móviles**

Octubre del 2015



# Contenido

Introducción .....	3
Descripción del problema .....	3
Objetivos .....	4
Motivación .....	5
Arquitectura de la aplicación .....	6
Esquema del diseño .....	6
Modelo de datos .....	16
Vistas .....	19
Conclusiones .....	30
Anexos .....	32
Listado de fuentes entregadas / Código fuente en GitHub .....	32
Manual de usuario .....	33

# Introducción

## Descripción del problema

En la empresa el proceso que se sigue para realizar un pedido es el siguiente:

- El cliente contacta con el comercial y hace su petición y negocian las condiciones.
- El comercial envía el pedido por diferentes medios a los administrativos.
- Los administrativos introducen los pedidos en el ERP.

Los comerciales pueden enviar los pedidos por teléfono o por correo electrónico, si lo envían por correo electrónico, pueden utilizar varios formatos, como enviar el pedido como cuerpo del mensaje, adjuntar una Excel con el pedido o enviar una foto del pedido (escrito a mano) por mail. Los administrativos del departamento de comercial, recogen los pedidos y los introducen en el ERP.

Con motivo de establecer una forma común y rápida de cumplimentar los pedidos por parte de los comerciales, se desarrolló un aplicación para el sistema Android, en la que se podían generar pedidos, esta aplicación se alimentaba tanto de los datos introducidos por el usuario, como de la base de datos de la empresa. El comercial introducía el pedido y se enviaba a los administrativos mediante un correo electrónico con un único formato.

La aplicación, para actualizar los datos necesitaba que existiera una conexión VPN con la empresa, ya que la empresa por motivos de seguridad no tiene accesibles los servidores desde Internet, por esta razón el comercial debía establecer una conexión VPN y luego actualizar los datos. A causa de esto la aplicación entro en desuso ya que era incomodo para los comerciales tener que actualizar los datos de esta forma.

## Objetivos

La aplicación constará de dos partes, la parte del servidor y la parte del cliente.

El servidor (Aplicación node.js):

- El servidor se encargara de gestionar los archivos a través del servicio de Google Drive, cuando los archivos se agreguen, modifiquen o eliminen, el servidor se encargara de actualizar estos con la carpeta de Drive. (Se hace necesaria esta utilidad ya que no existe un cliente oficial de Drive para servidores Linux).
- El servidor se encargara de gestionar los usuarios que tienen acceso a la carpeta mediante los permisos de Drive.
- El servidor se encargará de leer los correos enviados por el cliente, y mostrarlos de forma ordenada.

El cliente (Aplicación Android):

- El cliente tiene que solicitar los permisos adecuados para tener acceso a una cuenta de Drive, y comprobar que se tiene acceso a la carpeta compartida donde se ubican los ficheros CSV.
- El cliente tiene que obtener los ficheros de la cuenta Drive, y procesarlos para introducir los datos en la base de datos de la aplicación.
- El cliente tiene que gestionar la introducción, modificación y eliminación de los pedidos.
- El cliente se encargará de preparar y enviar el correo electrónico con los datos del pedido, preparados para la lectura por parte del servidor.
- El cliente se encargará de notificar cambios en la carpeta compartida.
- El cliente presentará información de los pedidos de forma resumida y fácilmente entendible.



## Motivación

Motivación personal surgida de una necesidad en la empresa.

Dado lo expuesto en el punto de descripción del problema, la motivación es realizar una herramienta que permita suplir, mediante servicios de almacenamiento en la nube, la conexión directa entre una aplicación cliente y su servidor. La aplicación realizada será una primera aproximación a una posible aplicación con mayores funcionalidades y herramientas, siendo está el punto de partida.

# Arquitectura de la aplicación

## Esquema del diseño

Casos de uso del Servidor:

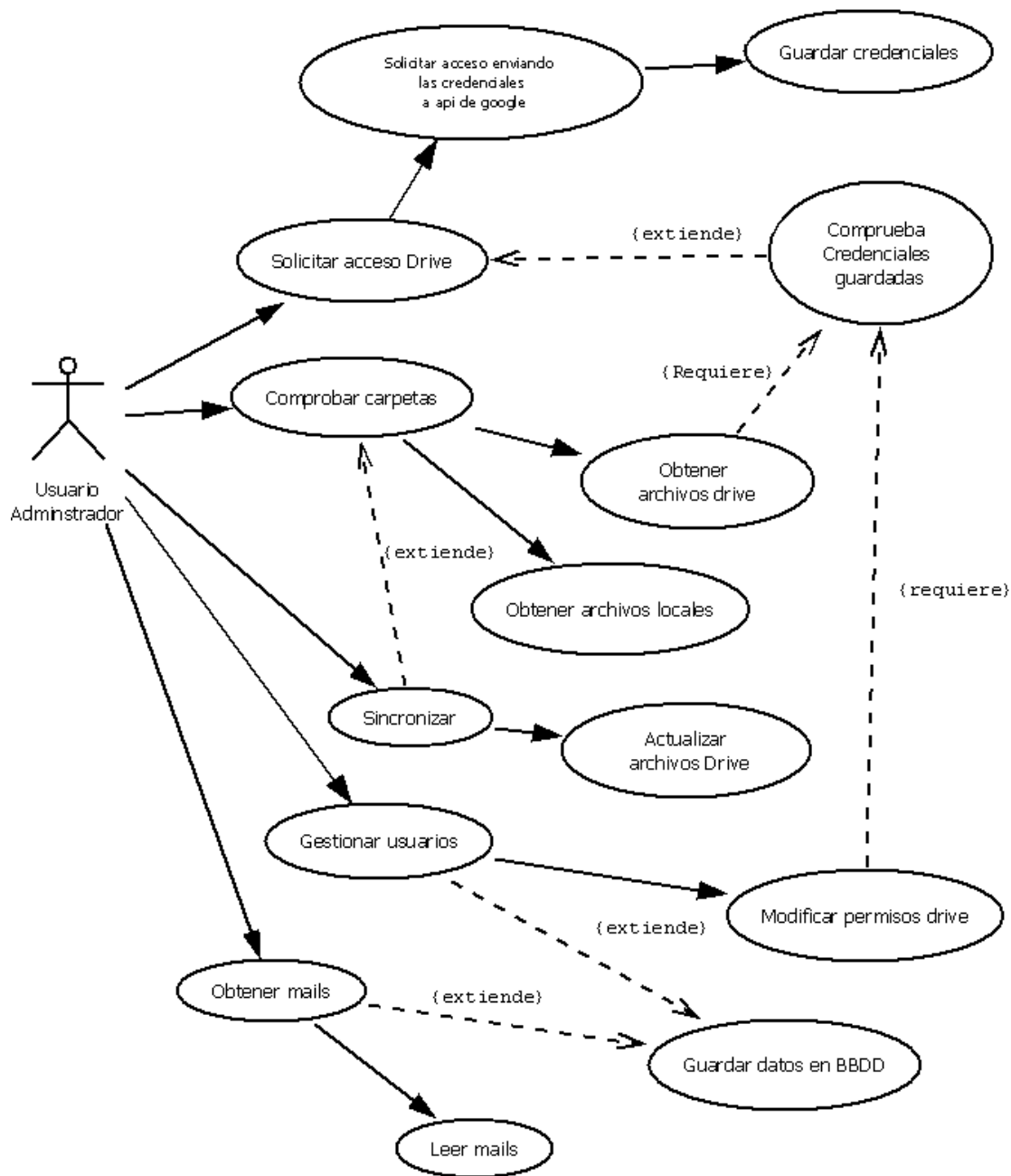


Fig. 1. Diagrama de casos de uso del servidor.

Casos de uso del Cliente:

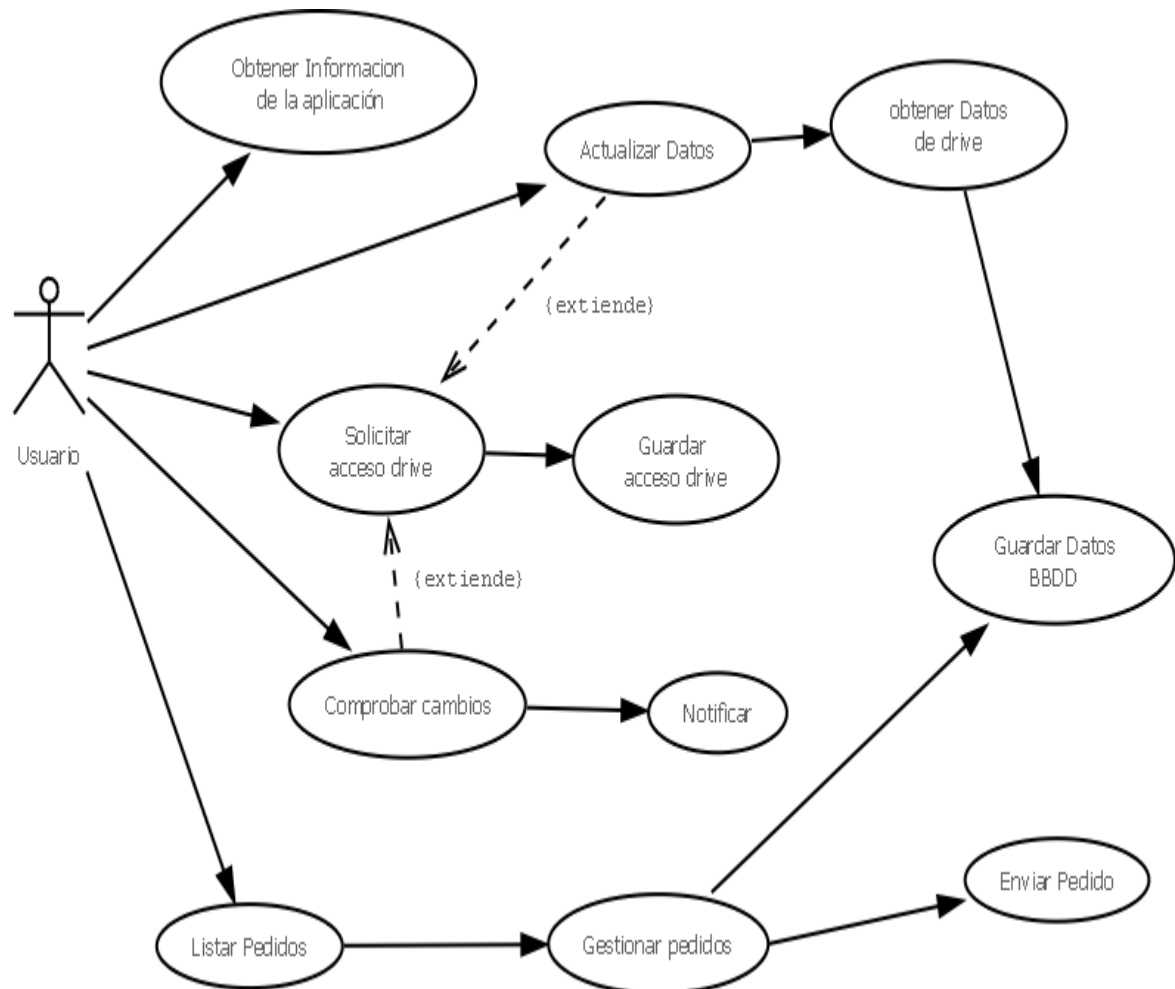


Fig. 2. Diagrama de casos de uso del cliente.

UML de la actividad principal:

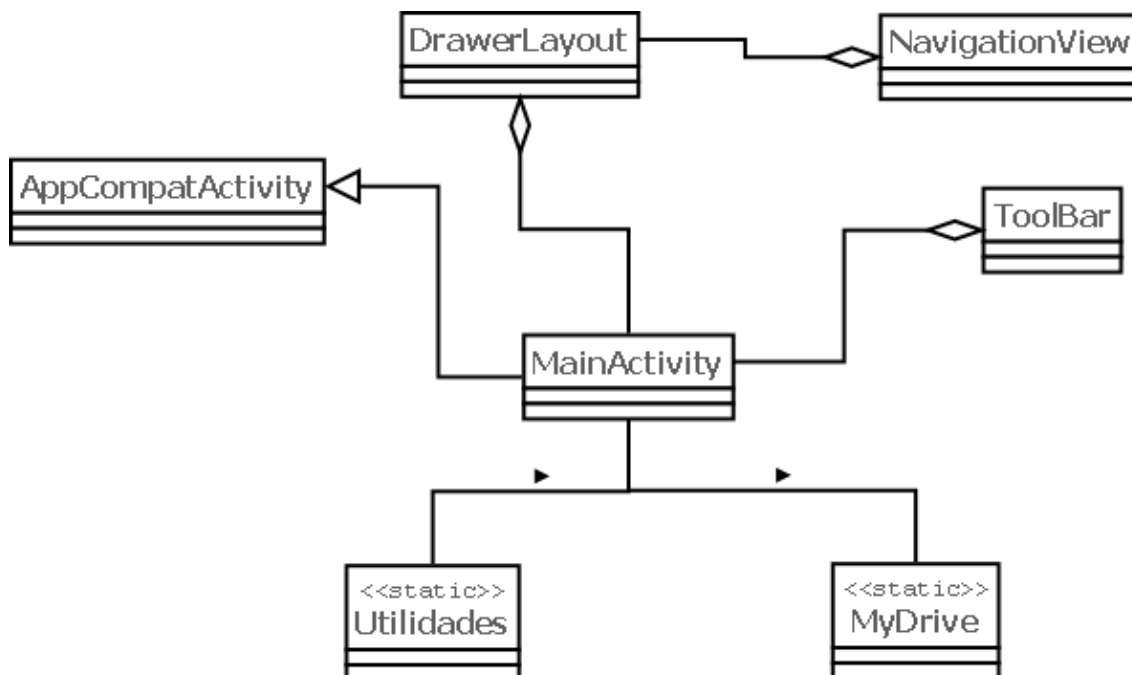


Fig. 3. UML de la actividad principal.

Se escoge extender de AppCompatActivity por las nuevas funcionalidades que soporta, de igual manera se utiliza el Toolbar, y el DrawerLayout con NavigationView, por las interfaces que aporta.

UML de las actividades información y configuración:

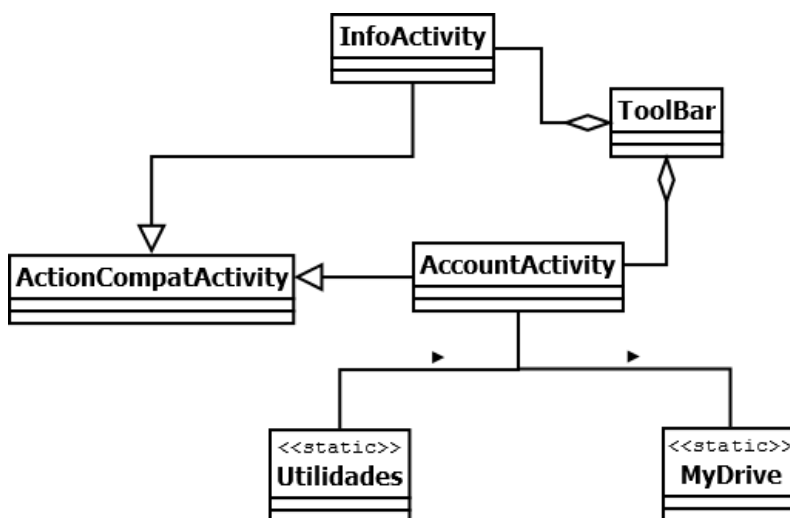


Fig. 4. UML de las actividades de configuración e información.



UML de la actividad pedidos:

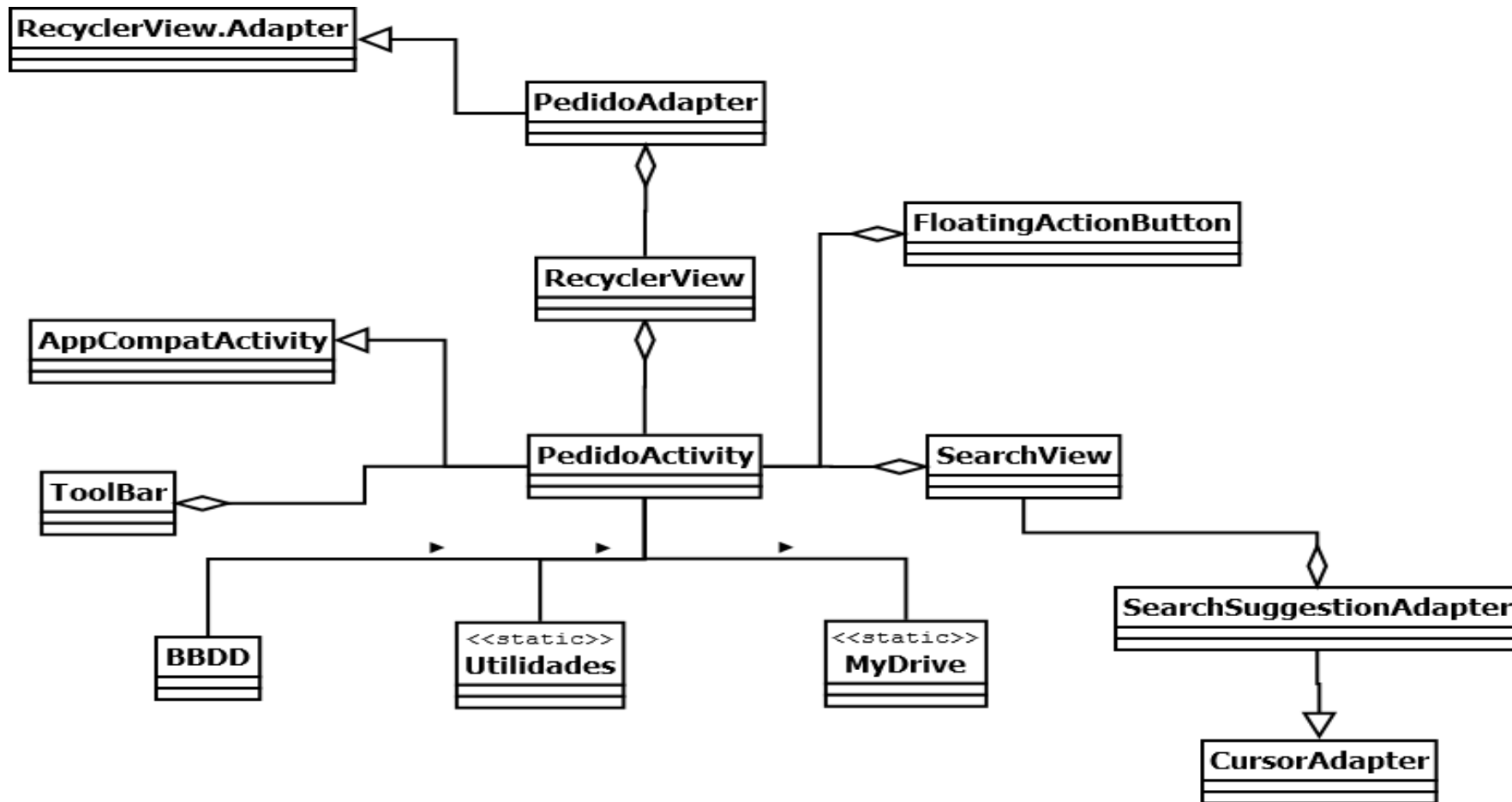


Fig. 5. UML de la actividad de pedidos.



Se decide utilizar el widget RecyclerView ya que el tamaño del listado de pedidos puede ser bastante extenso, y este elemento permite ir cargándolo dinámicamente sin saturar la aplicación.

Se decide utilizar el FloatingActionButton por su fácil adaptación a la interfaz, ya que no es necesario ocupar espacio en el Layout principal y actúa como cualquier botón.

Se decide utilizar el widget SearchView con el adaptador SearchSuggestionAdapter, que permite hacer búsquedas con autocompletado, se elige cuantos caracteres son necesarios para hacer la consulta, y hace un listado de posibles resultados, con la opción de poder devolver uno de ellos al seleccionarlo, El Adapter trabaja sobre un Cursor con los datos de la base de datos.

**Nota:**

A causa de una mal interacción con el FloatingActionButton cuando esta sobre un ListView o un RecyclerView, solo se usará si el dispositivo tiene una versión a partir de la 21. En otro caso se añadirá una opción al menú.

UML de la actividad detalle

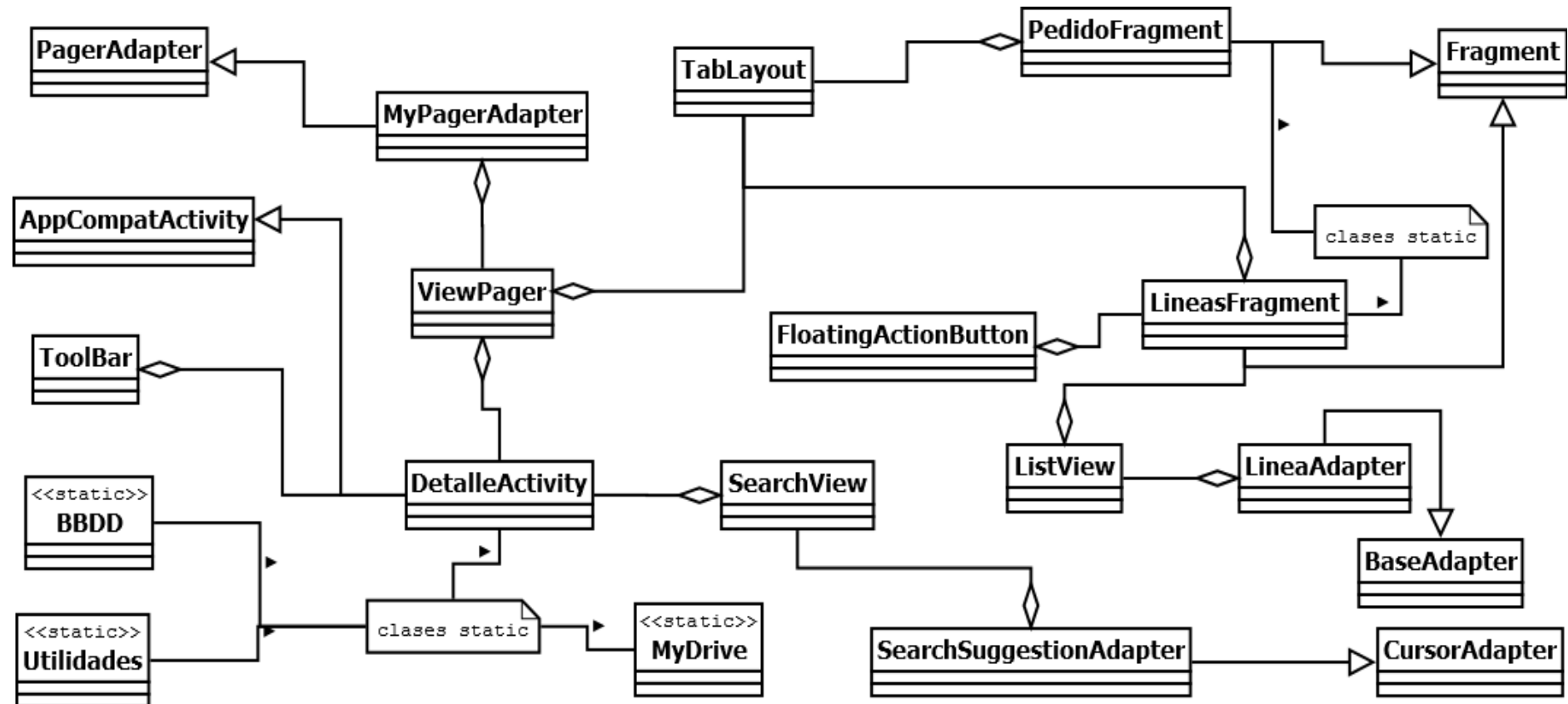


Fig. 6. UML de la actividad detalle.

Se decide utilizar un TabLayout para poder mostrar de forma fácil la cabecera del detalle y las líneas que lo forman. La cabecera como el cuerpo del pedido está formado por un Fragment.

**Nota:**

Por el problema con el FloatingActionButton se procede igual que en PedidoActivity.java.

UML de la actividad línea

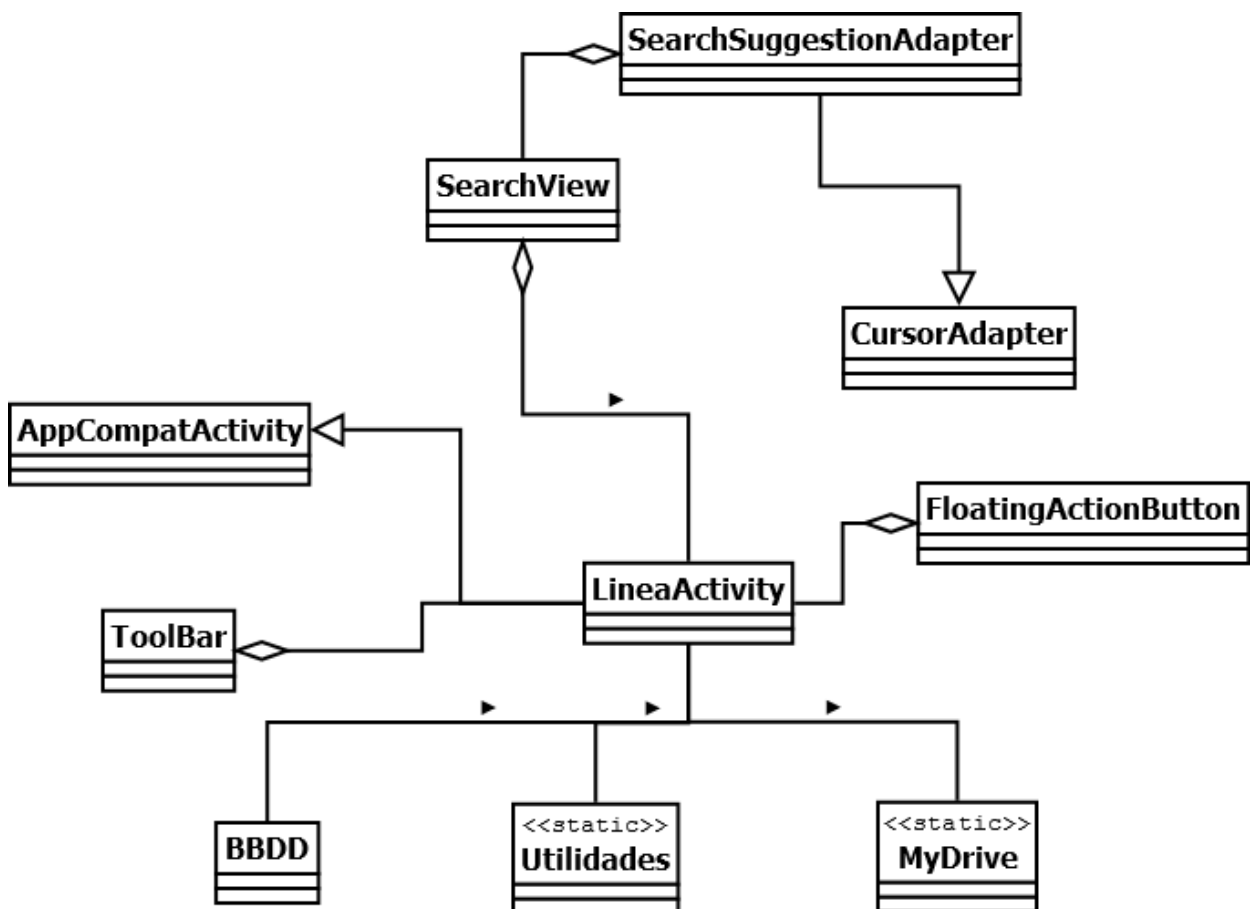


Fig. 7. UML de la actividad línea.

Procederé a realizar una breve explicación de los archivos generados.

### **Servidor**

El servidor se divide en una serie de módulos

- app.js  
Modulo principal de la aplicación, encargado de ejecutar el servidor, es el punto de entrada de las peticiones.
- logica.js  
Modulo encargado de generar las páginas de la aplicación web, así como de coordinar la utilización del resto de módulos, se encarga de comprobar y sincronizar los archivos locales con drive.
- drive.js  
Modulo encargado de proporcionar funciones para interactuar con drive, utiliza la API de Drive para gestionar las carpetas, archivos y permisos.
- mail.js  
Modulo encargado de gestionar el acceso y lectura del contenido de los correos electrónicos enviados por la aplicación cliente.
- bbdd.js  
Modulo encargado de la gestión de la base de datos.
- public/  
Todos los archivos necesarios para dar forma y funcionalidad a la aplicación web

### **Cliente**

El cliente se divide en una serie de paquetes

org.example.vicchiam.mispedidos

- MainActivity.java  
Actividad principal de la aplicación, se encarga de inicializar la base de datos, de inicializar las variables globales, y de comprobar el acceso a Drive al iniciar, también se encarga de lanzar el servicio que busca cambios en los archivos. Muestra la vista inicial de la aplicación.
- InfoActivity.java  
Actividad que se encarga de mostrar información sobre la aplicación.



- AccountActivity.java  
Actividad que se encarga de pedir los permisos de drive, y comprobar el acceso a la carpeta donde se alojan los archivos csv.
- PedidosActivity.java  
Actividad encargada de mostrar y gestionar los pedidos, tiene una utilidad para buscar los pedidos.
- DetalleActivity.java  
Actividad encargada de mostrar y gestionar las diferentes vistas que contiene, estas son PedidoFragment.java y LineasFragment.java
- LineaActivity.java  
Actividad encargada de mostrar y gestionar una línea del pedido, esta actividad permite buscar un producto y agregarlo al pedido.
- MyDriveService.java  
Implementa un servicio que busca cambios en el drive de los archivos indicados, se encuentra algún cambio lo notifica, para que el usuario actualice los datos.

#### org.example.vicchia.fragments

- PedidoFragment.java  
Fragment que permite mostrar y gestionar la cabecera de un pedido, tiene una herramienta de búsqueda con autocompletado y selección del cliente.
- LineasFragment.java  
Fragment que permite mostrar las líneas que tiene un pedido.

#### org.example.vicchiam.bbdd

- Pedido.java  
Clase que define un pedido, contiene un listado de Líneas
- Línea.java  
Clase que define una línea del pedido.
- Cliente.java  
Clase que define a un cliente.

- Producto.java  
Clase que define a un Producto
- DBPedido.java  
Clase que agrupa las operaciones de consulta, creación, modificación y borrado de pedidos y líneas en la base de datos.
- DBCliente.java  
Clase que agrupa las operaciones de consulta, creación, modificación y borrado de clientes en la base de datos.
- DBProducto.java  
Clase que agrupa las operaciones de consulta, creación, modificación y borrado de productos en la base de datos.

#### org.example.vicchiam.adapters

- SearchSuggestionAdapter.java  
Clase que se encarga de dar forma al autocompletado en las diferentes búsquedas.
- PedidosAdapter.java  
Clase que se encarga de representar los ítems del RecyclerView de la actividad PedidosActivity.java

#### org.example.vicchiam.utilidades

- Utilidades.java  
Clase que contiene diversos métodos estáticos como cargar guardar las preferencias, método para comprobar el estado de la red, método para transformar fechas, etc.
- MyDrive.java  
Clase que contiene varios atributos y métodos estáticos para la gestión de todo lo relacionado con Drive, obtener credenciales, obtener un fichero, buscar una carpeta, etc.
- BBDD.java  
Clase que contiene varios atributos estáticos que contienen objetos de acceso a la base de datos.
- PedidoTEMP.java  
Clase que contiene un atributo y un método estáticos utilizados para la creación y modificación de pedidos

## Modelo de datos

Diagrama ER de la base de datos del Servidor:

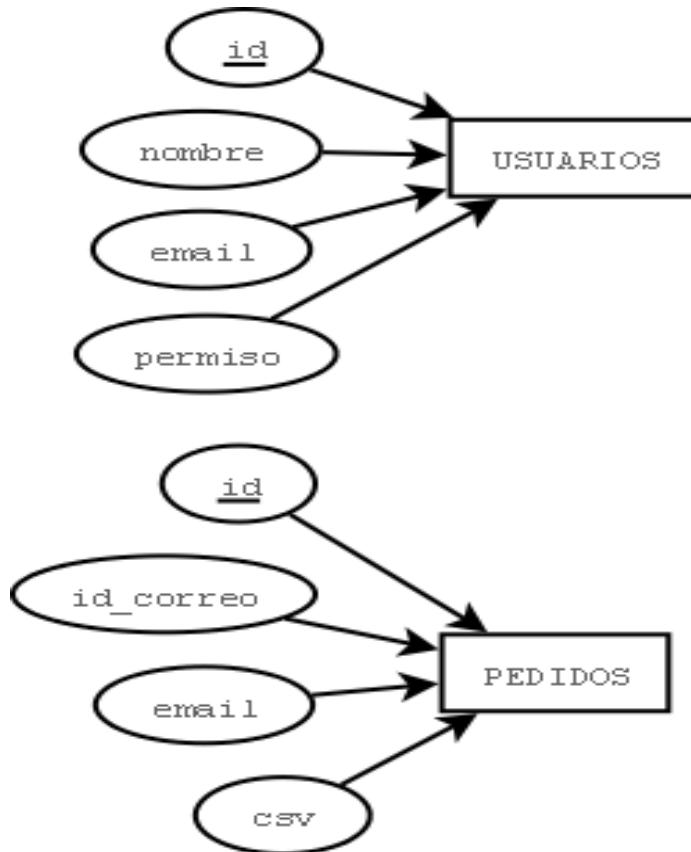
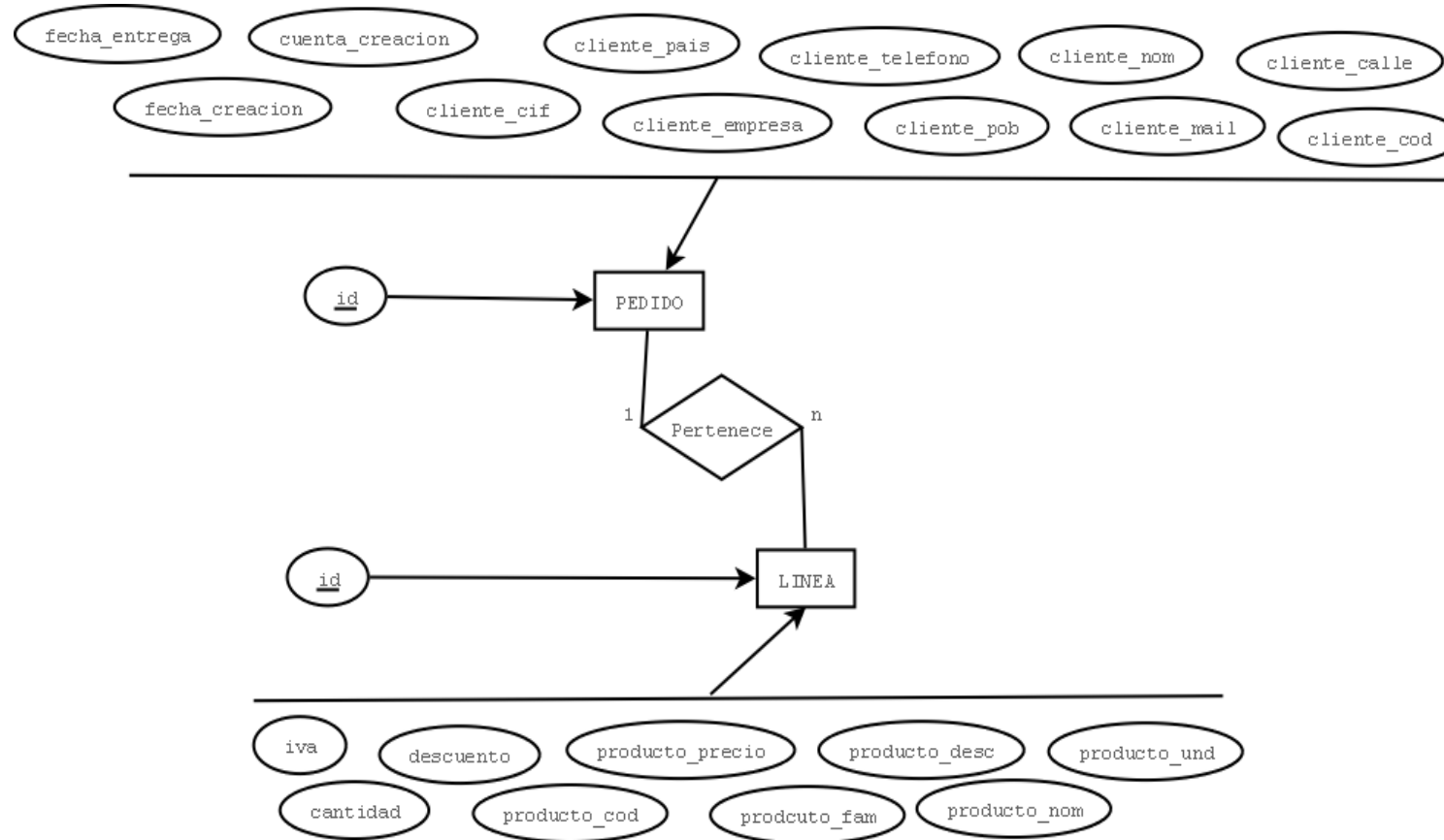


Fig. 8. Modelo de la base de datos del servidor

Dos tablas no relacionadas que contendrán los datos necesarios para gestionar los clientes y los mails recibidos.



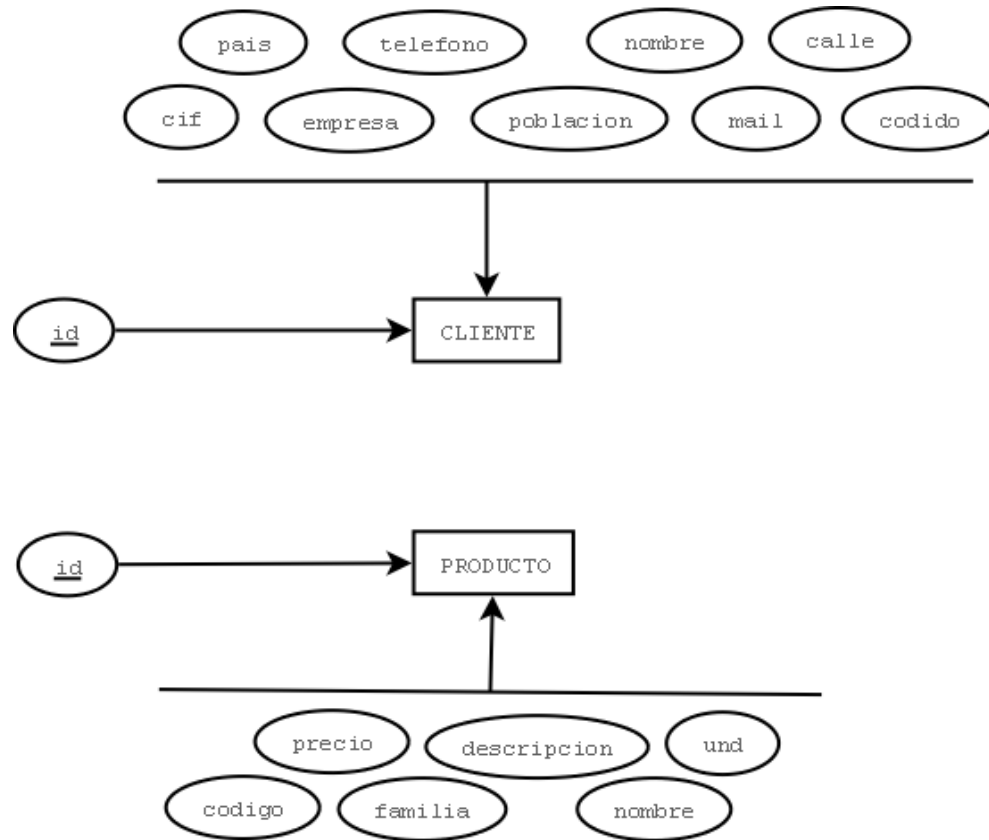
Diagrama ER de la base de datos del cliente 1



Tablas pedido y líneas relacionadas, ya que la tabla Pedido representa la cabecera del pedido, y la tabla Línea, cada uno de los productos que compone el pedido.

Fig. 9. Modelo ER de la base de datos del cliente (parte 1)

Diagrama ER de la base de datos del cliente 2



Se ha decidido no relacionar estas tablas con la tabla pedidos, ya que al ser estos los datos que se importan de los CSV, y cada nueva importación se borran completamente las tablas y sus índices. Al no asegurarse de que los índices serán los mismos, y por posibles cambios en las fuentes de datos, las propiedades más relevantes del cliente o producto pasarán a formar parte de la tabla pedidos y líneas respectivamente, así los pedidos permanecerán inamovibles a los cambios en las posteriores importaciones, salvo su expresa modificación.

Fig. 10. Modelo ER de la base de datos del cliente (parte 2)

## Vistas

Esquema que muestre las principales pantallas de la aplicación y el diagrama de navegación.

Diagramas de navegación:

Nomenclatura utilizada:

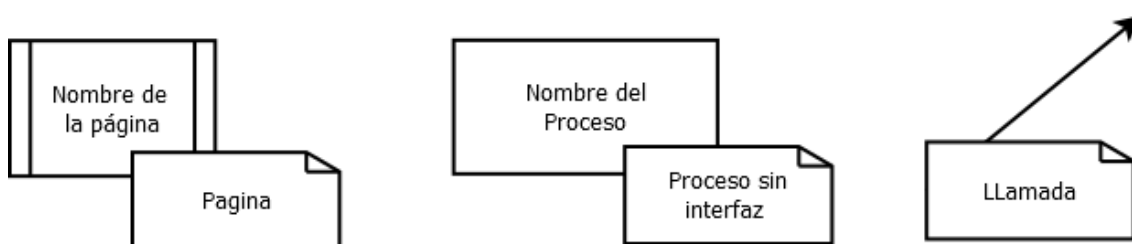


Fig. 11. Nomenclatura

Diagrama de navegación del Servidor:

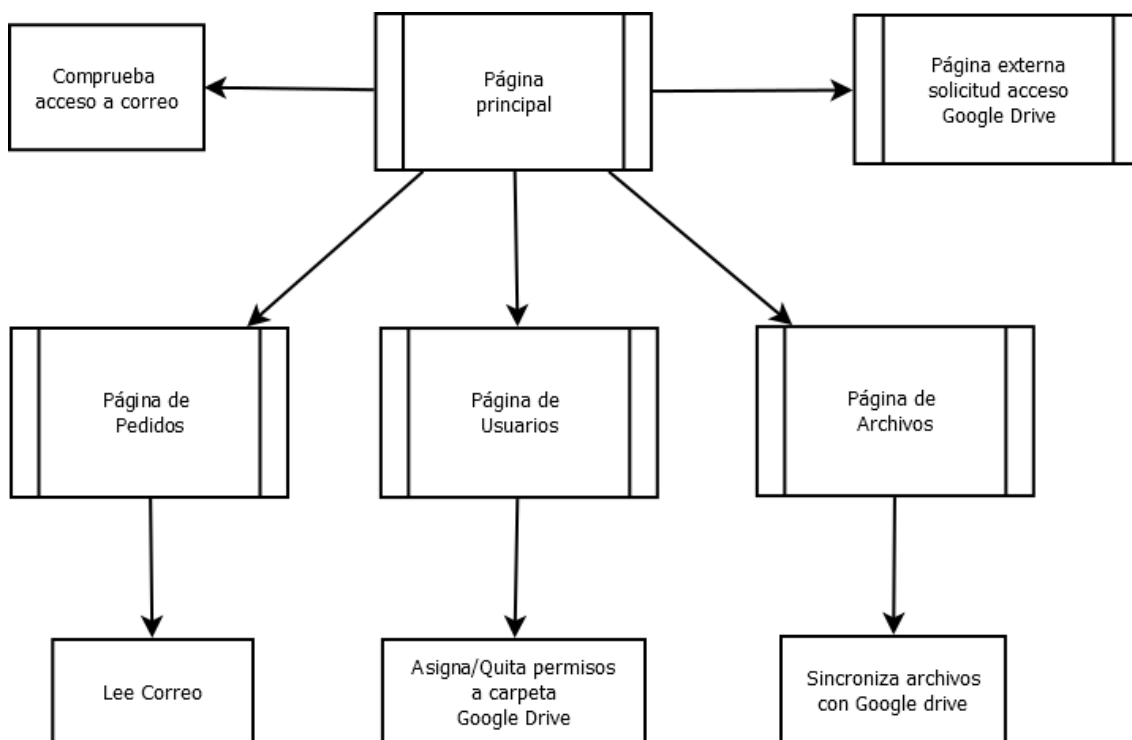


Fig. 12. Diagrama de navegación del servidor.

Diagrama de navegación del Cliente:

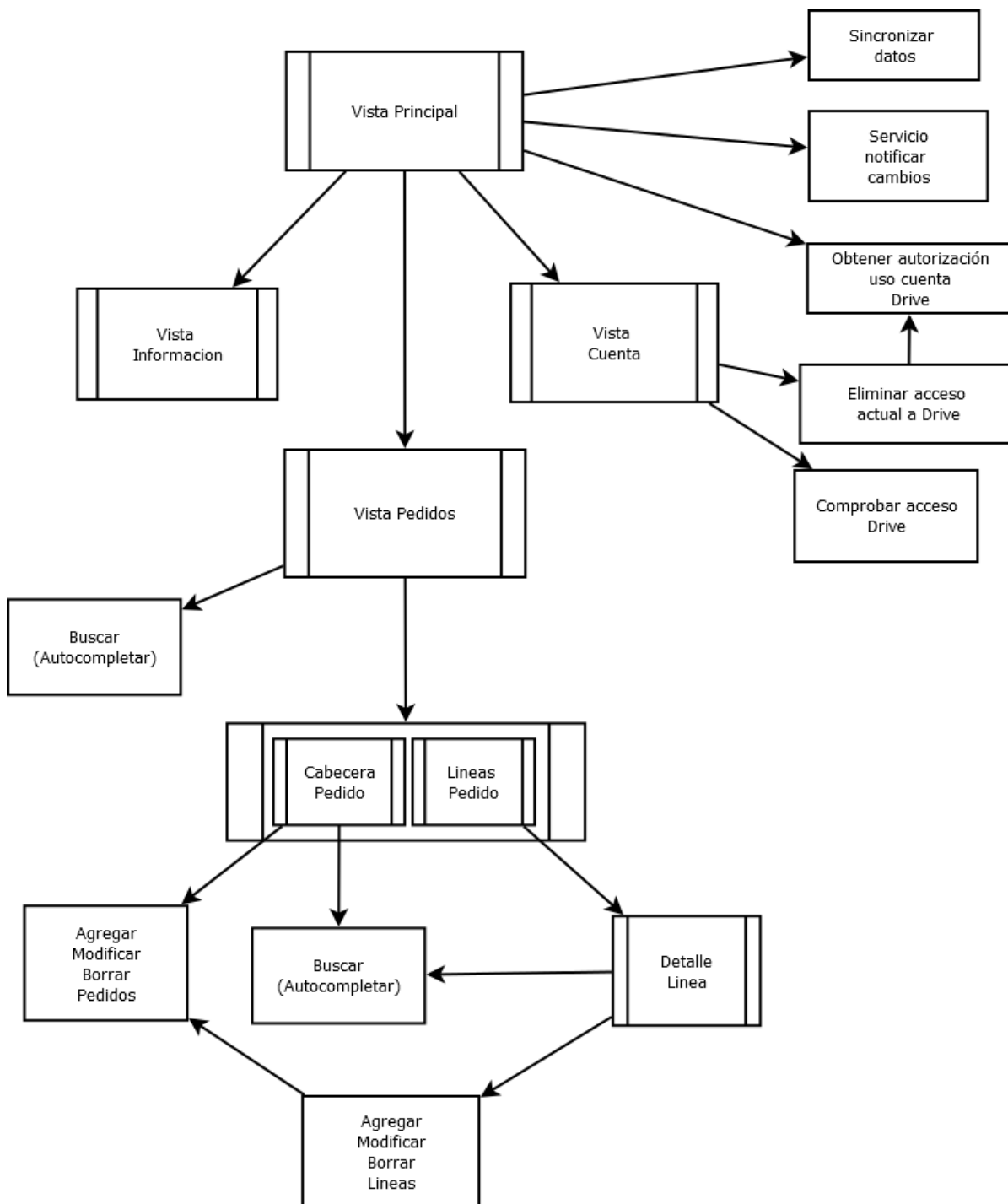


Fig. 13. Diagrama de navegación del cliente.

Vistas Servidor:

Página Inicial, lista los pedidos.

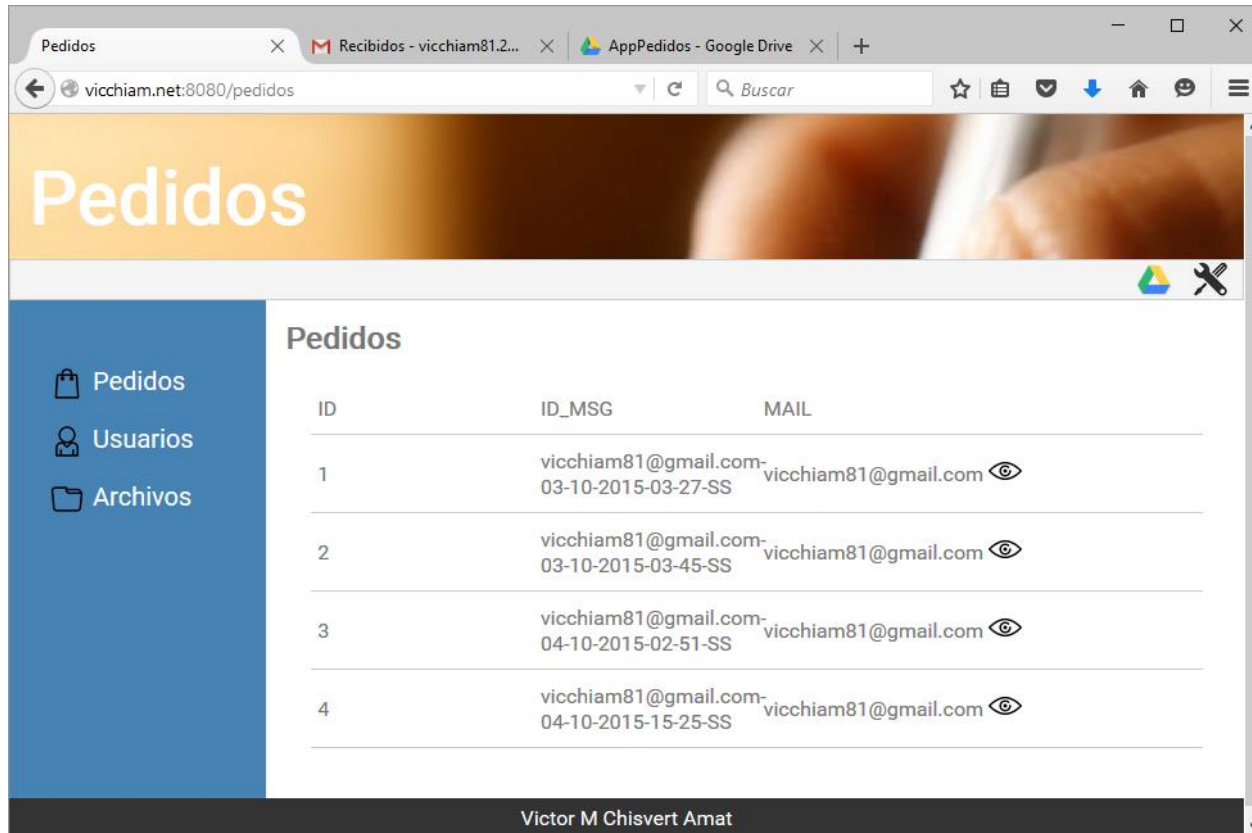


Fig. 14. Página inicial del Servidor



Detalle de un pedido.

**Pedido**

Cliente: Alabama Educational Tv Comm  
Contacto: Sarah Candlish  
Mail: sarah.candlish@gmail.com  
Tel: 770-732-1194  
Direccion: 45 2nd Ave #9759 Atlanta

Producto	Precio Und	Cantidad	Desc	Iva	Total
Pantone Reference Library	1623	2 und	20%	10%	2856 Euros

Victor M Chisvert Amat

Fig. 15. Detalle del pedido.

## Gestión de usuarios

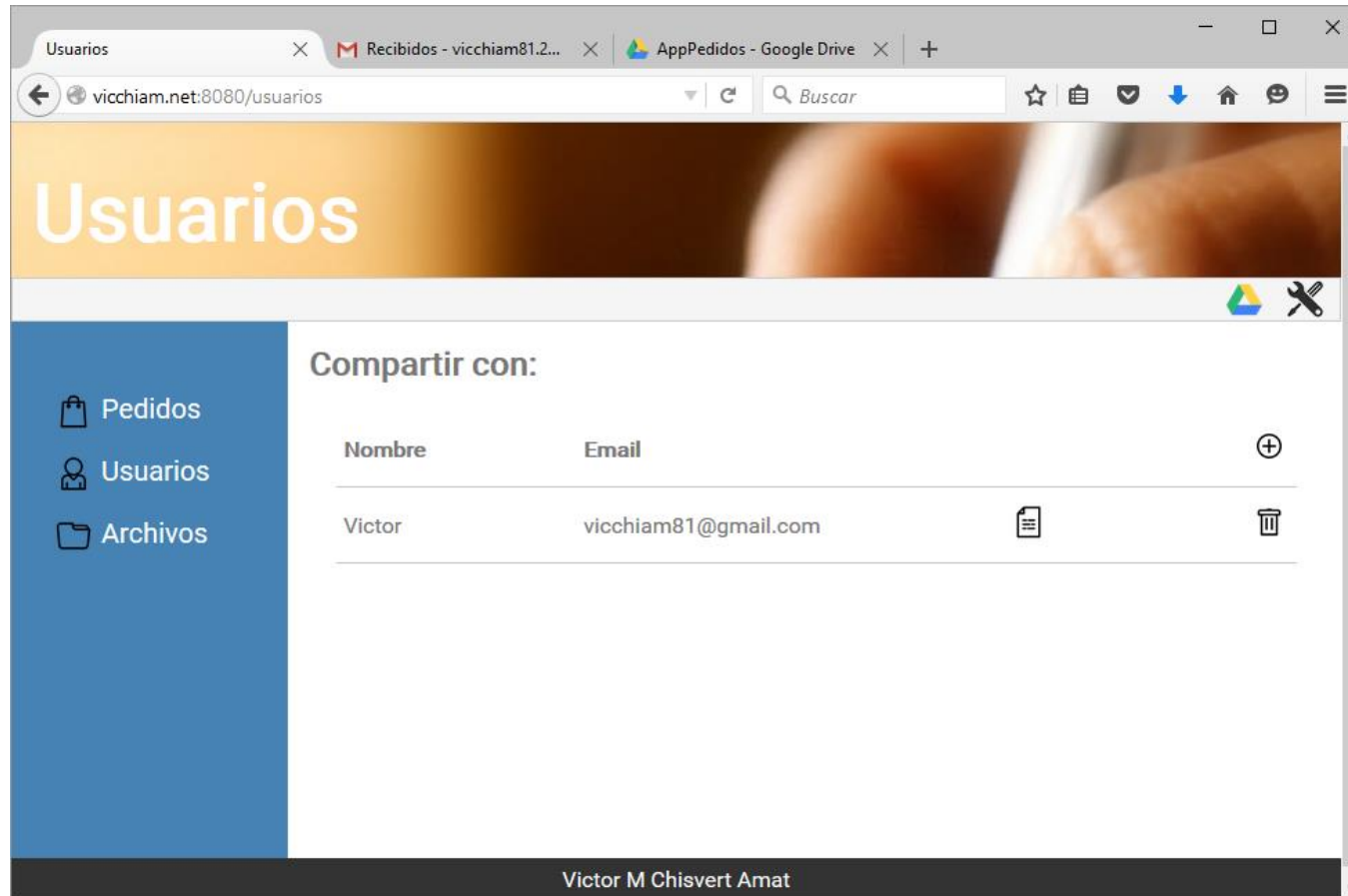
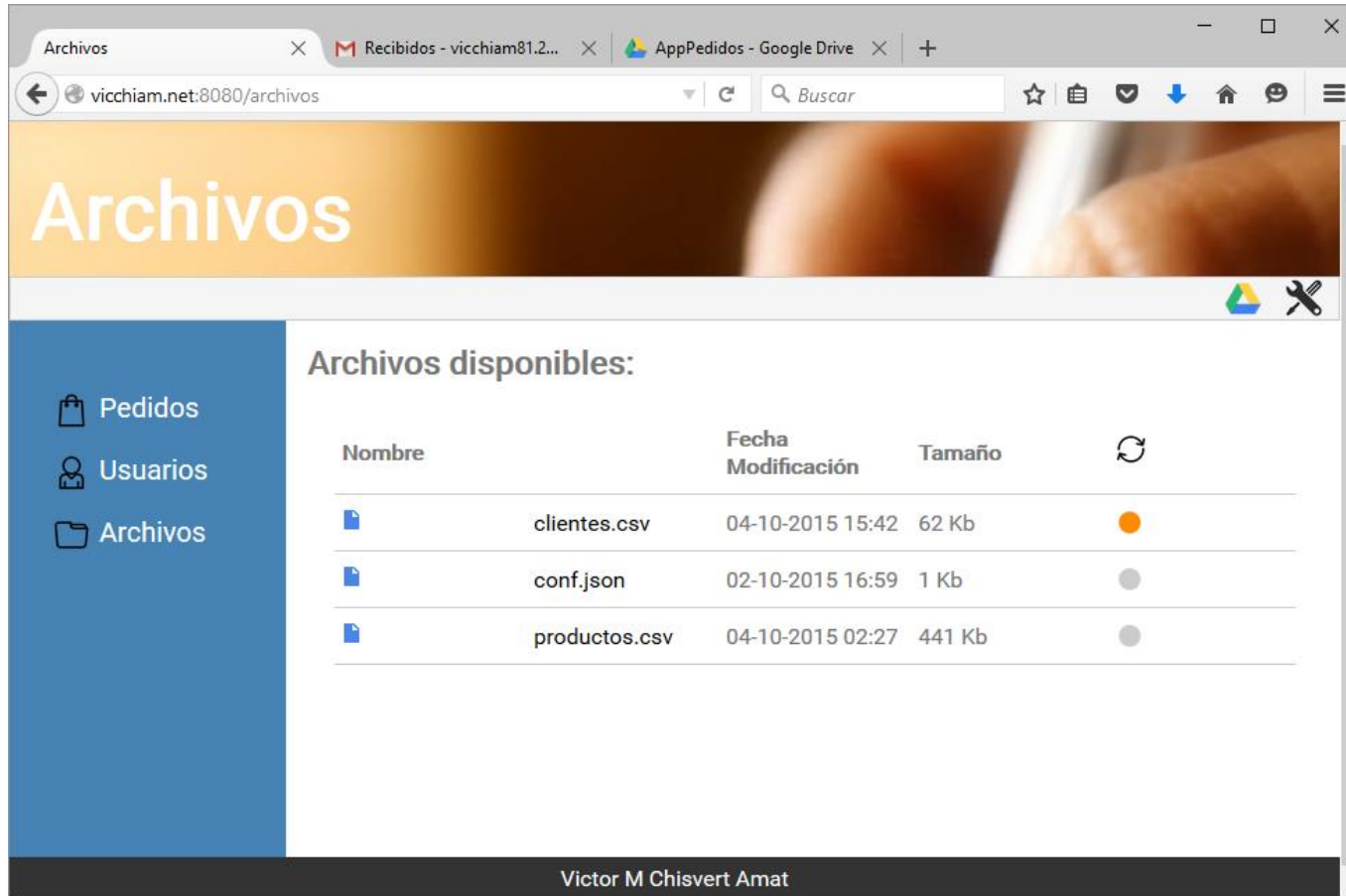


Fig. 16. Gestión de usuarios.

## Gestión de archivos



Archivos disponibles:

Nombre	Fecha Modificación	Tamaño	
clientes.csv	04-10-2015 15:42	62 Kb	●
conf.json	02-10-2015 16:59	1 Kb	●
productos.csv	04-10-2015 02:27	441 Kb	●

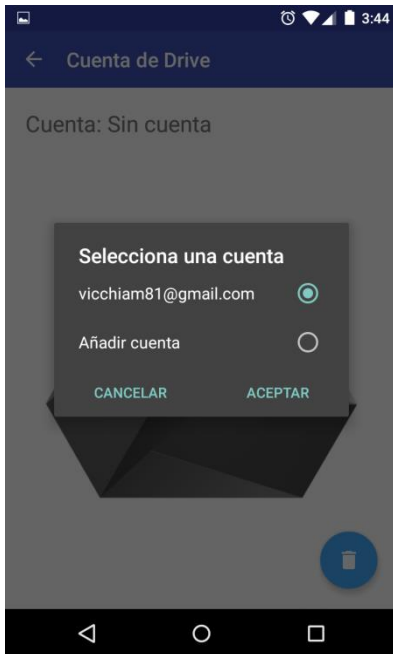
Victor M Chisvert Amat

Fig. 17. Gestión de Archivos

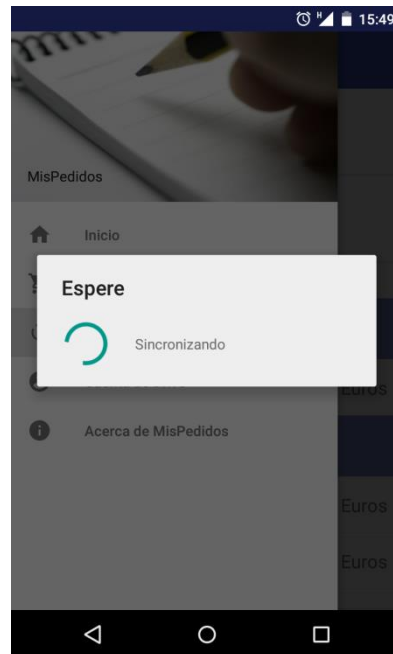


## Vistas Cliente

Primer inicio de la aplicación se agrega la cuenta para acceder a Drive y se sincronizan los datos por primera vez.

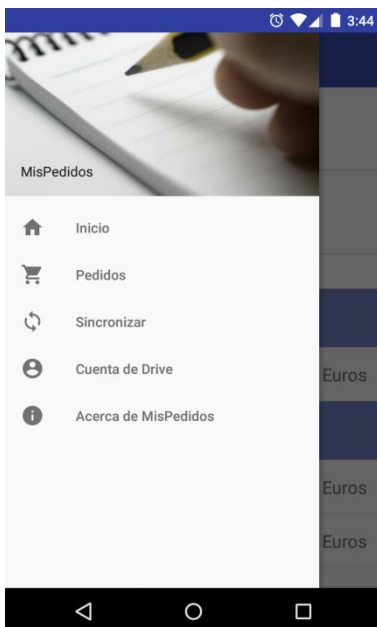


**Fig. 18. Agregar cuenta**



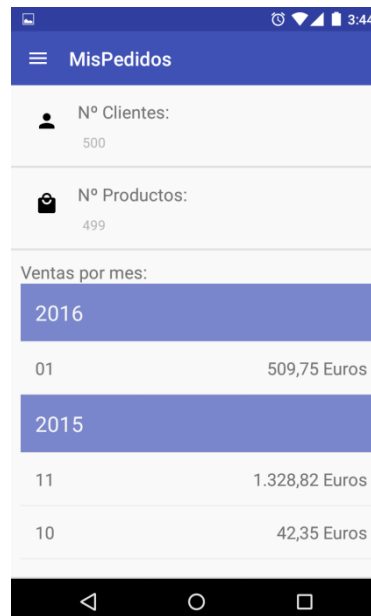
**Fig. 19. Sincronizar datos**

Inicio normal de la aplicación.



**Fig. 20. Vista inicial menú abierto**

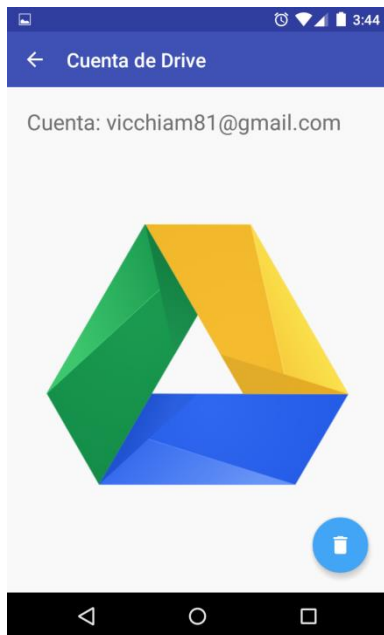
Muestra algunos datos resumidos



**Fig. 21. Vista inicial menú cerrado**

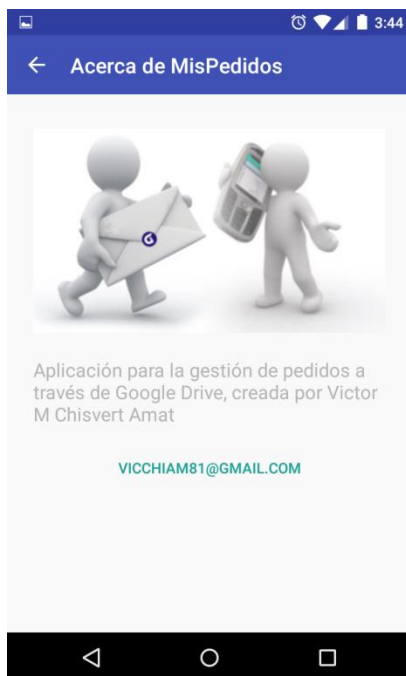


Vista cuenta, comprueba el acceso a Drive, se puede cambiar la cuenta de acceso.



**Fig. 22. Vista cuenta**

Vista Información



**Fig. 23. Información de la aplicación.**

Listado de pedidos



Fig. 24. Pedidos

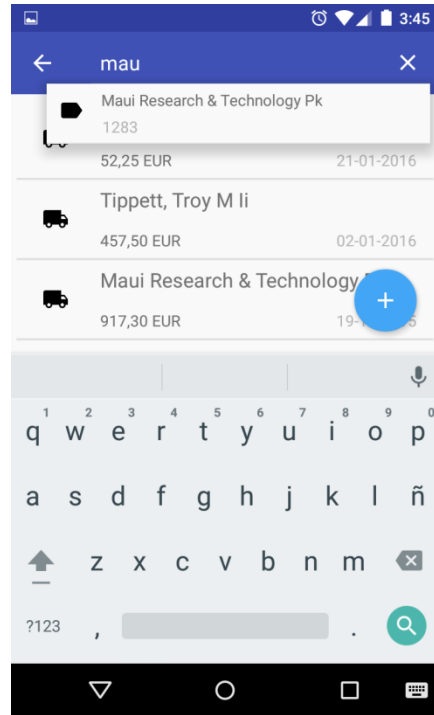


Fig. 25. Buscar pedidos

Gestión Pedidos

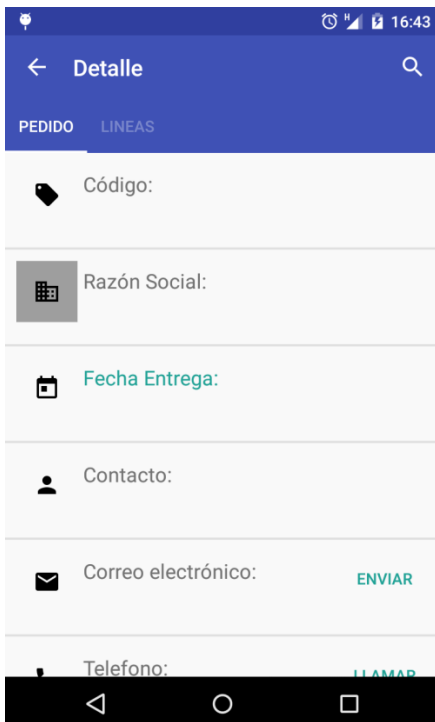


Fig. 26. Gestión

Nuevo pedido (Buscar)

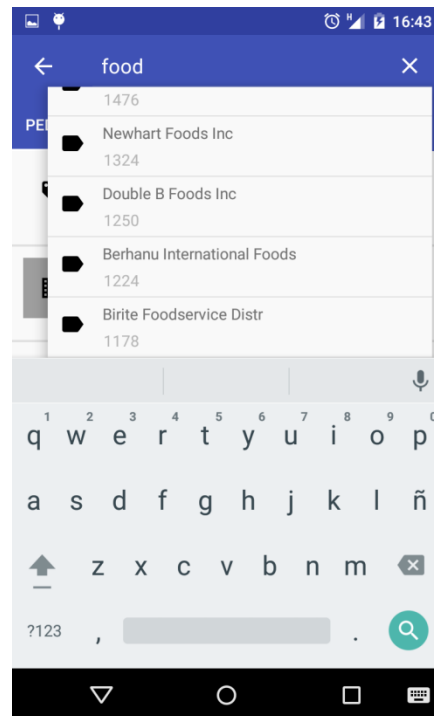


Fig. 27. Nuevo pedido (Buscar)

### Gestión de líneas

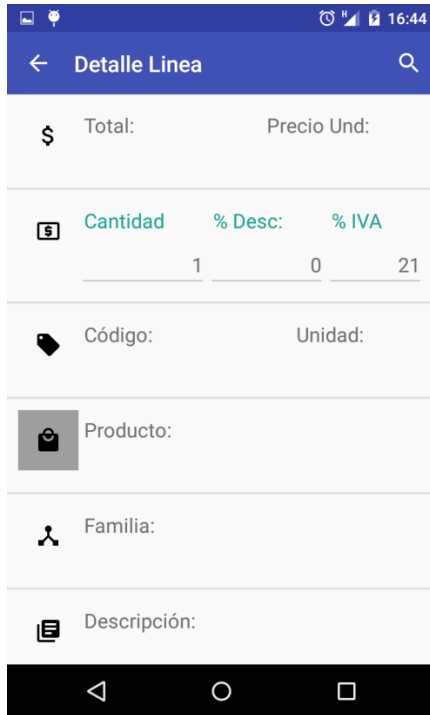


Fig. 28. Gestión líneas

### Nueva línea (buscar)

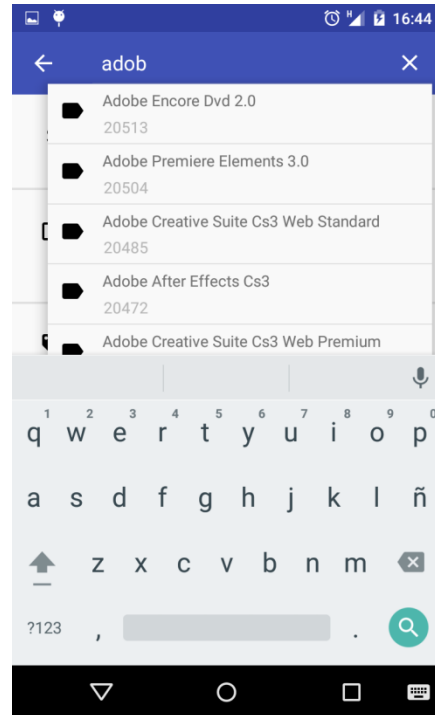


Fig. 29. Nueva línea (buscar)

### Líneas

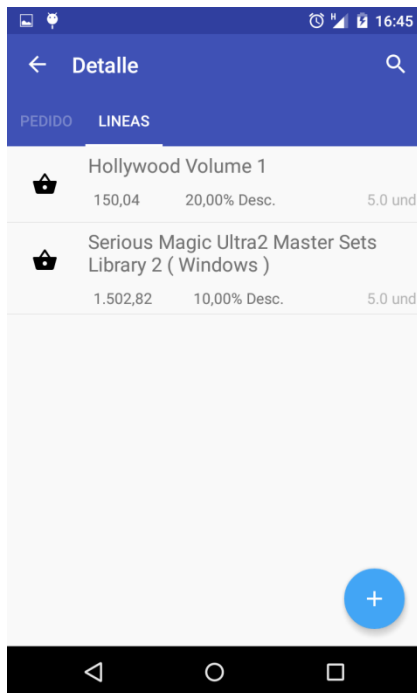


Fig. 30. Líneas

Interfaz de pedidos en Android API < 21



Fig. 31. Vista pedidos Android API <21

Interfaz de líneas en Android API <21



Fig. 32. Vista líneas en Android API < 21

## Conclusiones

El grado de cumplimiento de los objetivos en el servidor son:

El servidor es capaz de pedir permiso para acceder a la cuenta de Drive y crear la carpeta que utilizará con posterioridad.

El servidor es capaz de sincronizar con la carpeta de drive los archivos que contiene en su carpeta local, agregando, actualizando o eliminando estos de la carpeta d si es necesario.

Tiene la capacidad de agregar o eliminar usuarios, modificando en Drive los permisos que tiene estos sobre la carpeta compartida.

Tienen la capacidad de comprobar el acceso a una cuenta de correo, donde se leerán los pedidos, y si es correcto guardar las propiedades de conexión, y generar un archivo de configuración para compartir la cuenta con los clientes.

Puede leer la cuenta de correo indicada para descargar y leer los adjuntos enviados por los clientes, guardando estos en la base de datos, y mostrando un listado y detalle de estos.

Las líneas abiertas en el servidor son:

Capacidad para utilizar otros servicios de almacenamiento, como Dropbox.

Capacidad de generar los archivos de una forma personalizada para cada cliente.

Capacidad de generar los CSV conectando directamente con la base de datos que contiene los datos.

Gestionar los pedidos, estados de pedidos (recibido, pendiente), eliminar pedidos, etc.

Mejorar la interfaz que muestra el detalle del pedido.

Automatizar el proceso de sincronización.

Para la lectura de correos de la cuenta Gmail, se debe permitir que se acceda a esta a través de un IMAP con seguridad TLS, se debe indicar expresamente.



El grado de cumplimiento de los objetivos en el cliente son:

El cliente es capaz de pedir y obtener acceso a la cuenta de Drive indicada por el cliente, como comprobar el acceso a la carpeta de la aplicación de Drive.

El cliente es capaz de obtener los CSV indicados, y guardarlos en la base de datos de la aplicación.

El cliente puede obtener la cuenta de correo donde se deberán mandar los pedidos desde el fichero de configuración alijado en Drive.

El cliente tiene la capacidad de agregar, modificar y eliminar pedidos, tiene la utilidad de buscar pedidos con autocompletado.

El cliente es capaz de buscar y mostrar los clientes disponibles de la base de datos de una forma fácil, y agregarlos a la cabecera de un pedido.

El cliente es capaz de agregar, modificar y eliminar líneas de pedidos.

El cliente puede buscar y mostrar un listado de productos de una forma fácil y rápida.

El cliente puede generar un correo electrónico debidamente formateado para enviar al servidor.

El cliente puede notificar si ha habido algún cambio en la carpeta de la aplicación.

Las líneas abiertas en el cliente son:

La capacidad de utilizar otros servicios de almacenamiento como Dropbox.

La capacidad de obtener los CSV personalizados para la cuenta del cliente.

## Anexos

### Listado de fuentes entregadas / Código fuente en GitHub

Fuentes entregados:

Cliente Cliente.zip

Servidor Servidor.zip

GitHub:

Servidor:

<https://github.com/vicchiam/pedidos>

Cliente:

<https://github.com/vicchiam/AndroidMisPedidos>

Youtube:

Servidor:

<https://www.youtube.com/watch?v=M5fXgzUOVQU>

Cliente:

[https://www.youtube.com/watch?v=DyQjVaV\\_JRo](https://www.youtube.com/watch?v=DyQjVaV_JRo)



## Manual de usuario

Manual del Servidor:

### Preparar los archivos a sincronizar por el servidor:

Se tienen que crear dos archivos CSV.

clientes.csv

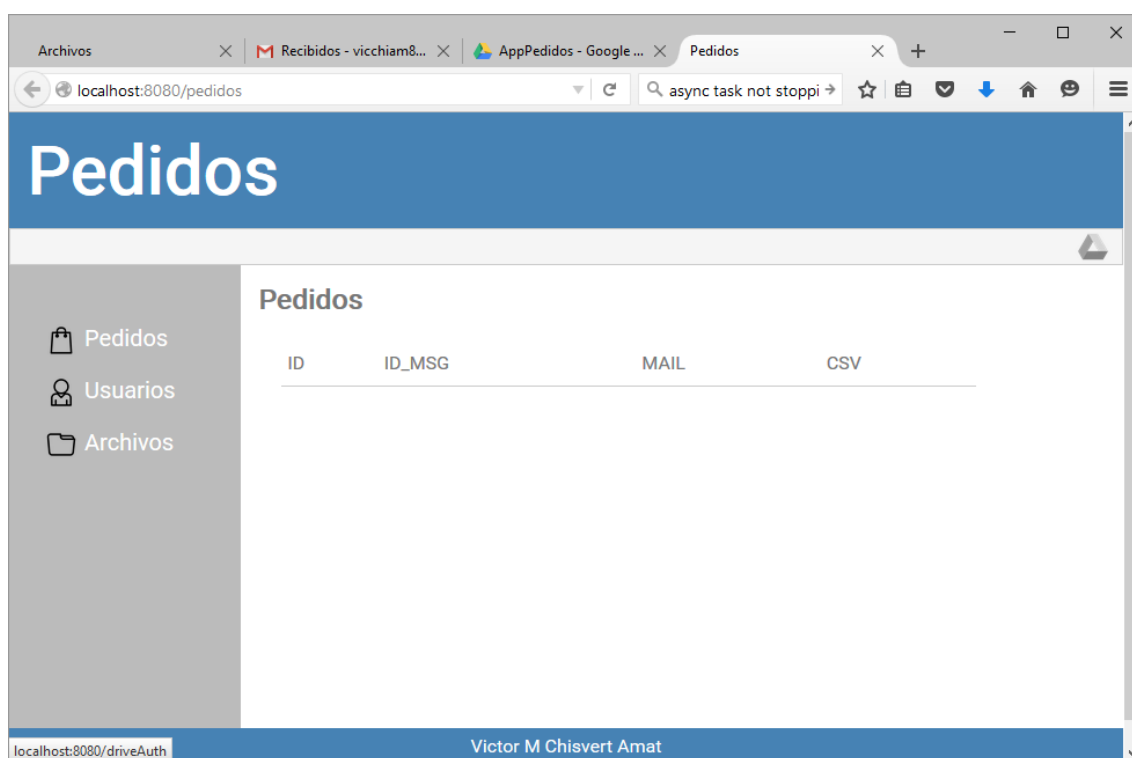
CODIGO;CIF;NOMBRE;TELEFONO;MAIL;CALLE;POBLACION;CP;PAIS;EMPRESA

productos.csv

CODIGO;NOMBRE;FAMILIA;UND;PRECIO;DESCRIPCION

### Inicio de la aplicación.

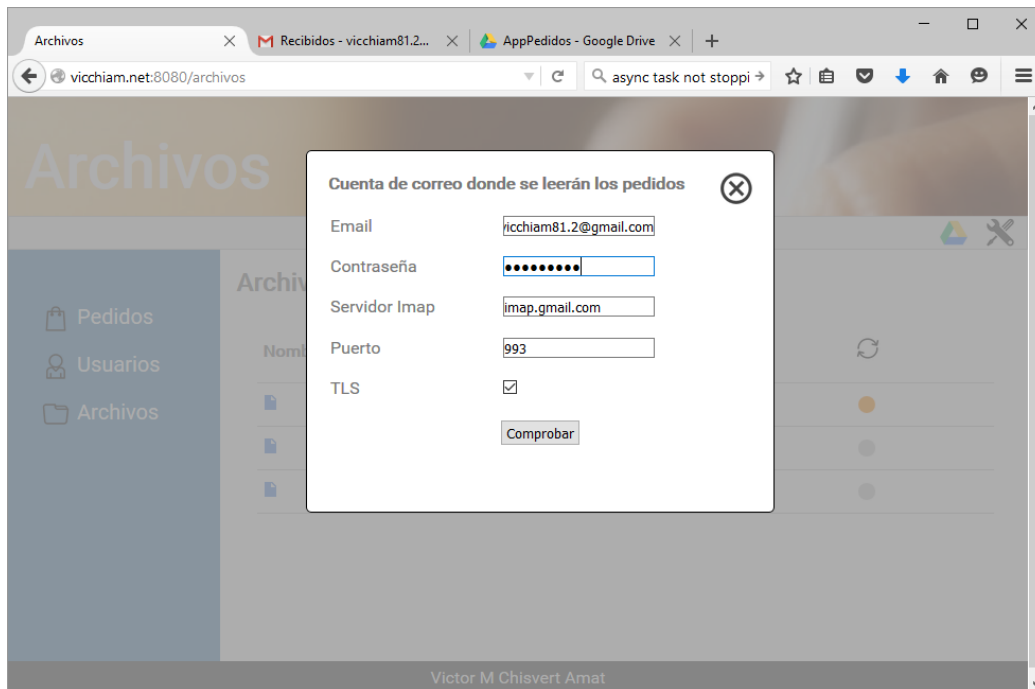
El icono de Drive de la derecha esta gris, eso significa que no hay una cuenta asociada, se deberá clicar sobre este icono para obtener los permisos necesarios.



Una vez obtenido los permisos el sistema procederá a crear la carpeta necesaria AppPedidos.

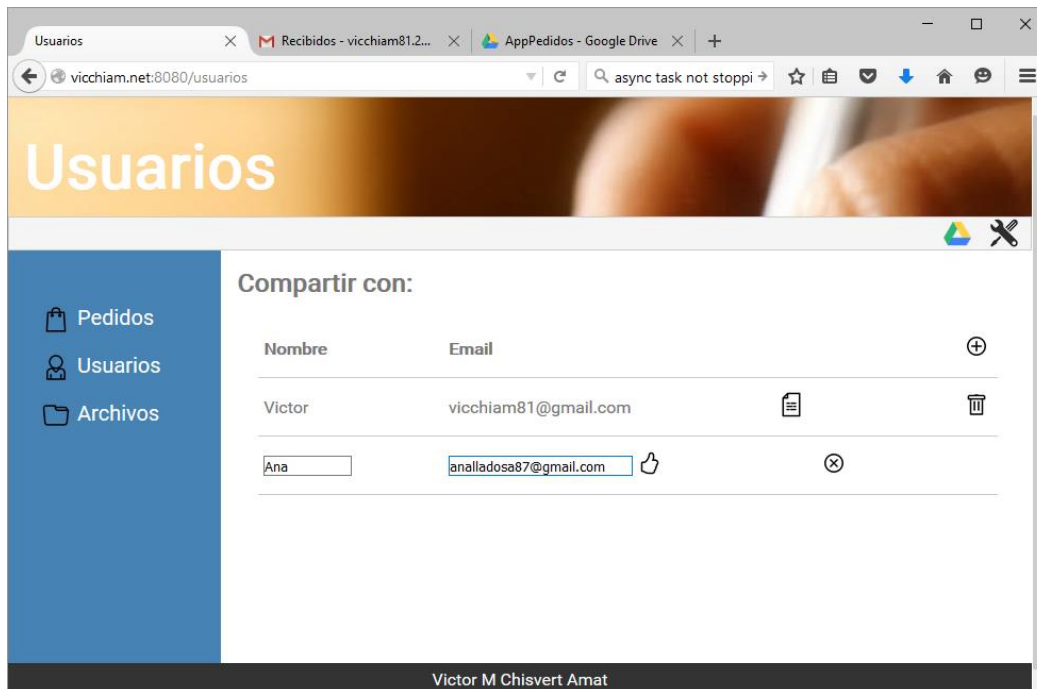


Obtener la cuenta de correo donde los clientes enviarán los correos



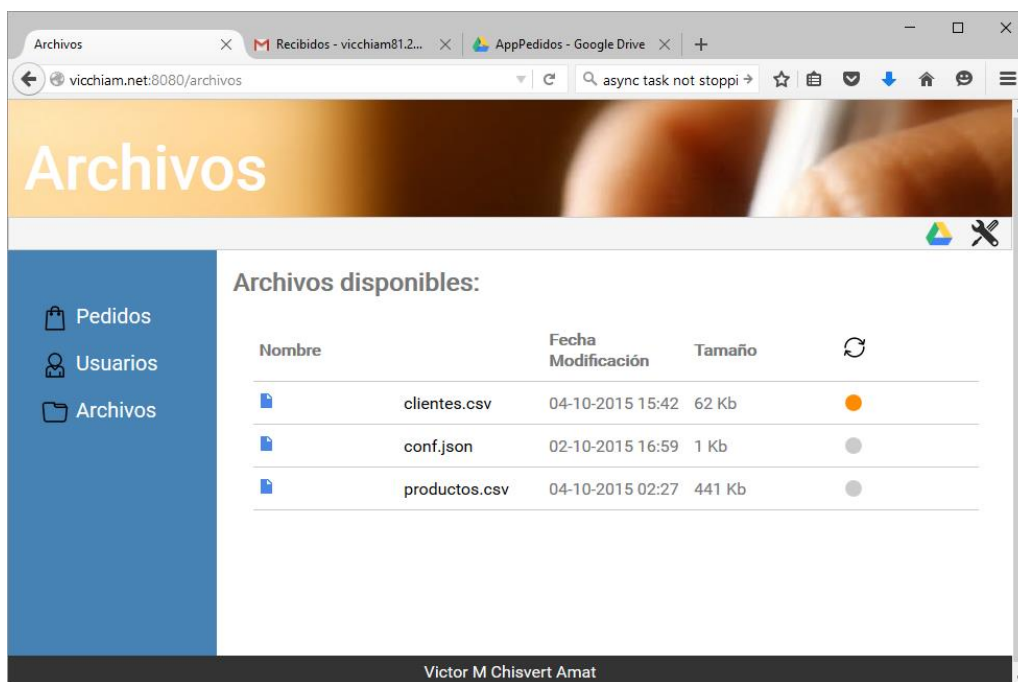
Pulsando el icono de herramientas, se abrirá una ventana donde deberás indicar los parámetros de conexión, que si accede correctamente a la cuenta, generara o modificará un archivo de configuración que leerán los clientes.

### Agregar usuarios.



Se podrán agregar y eliminar usuarios que pueden acceder a la carpeta compartida de Drive, a través de esta interfaz. Una vez agregado el usuario recibirá un correo en su cuenta Gmail indicando que se ha compartido una carpeta con él, este deberá aceptar.

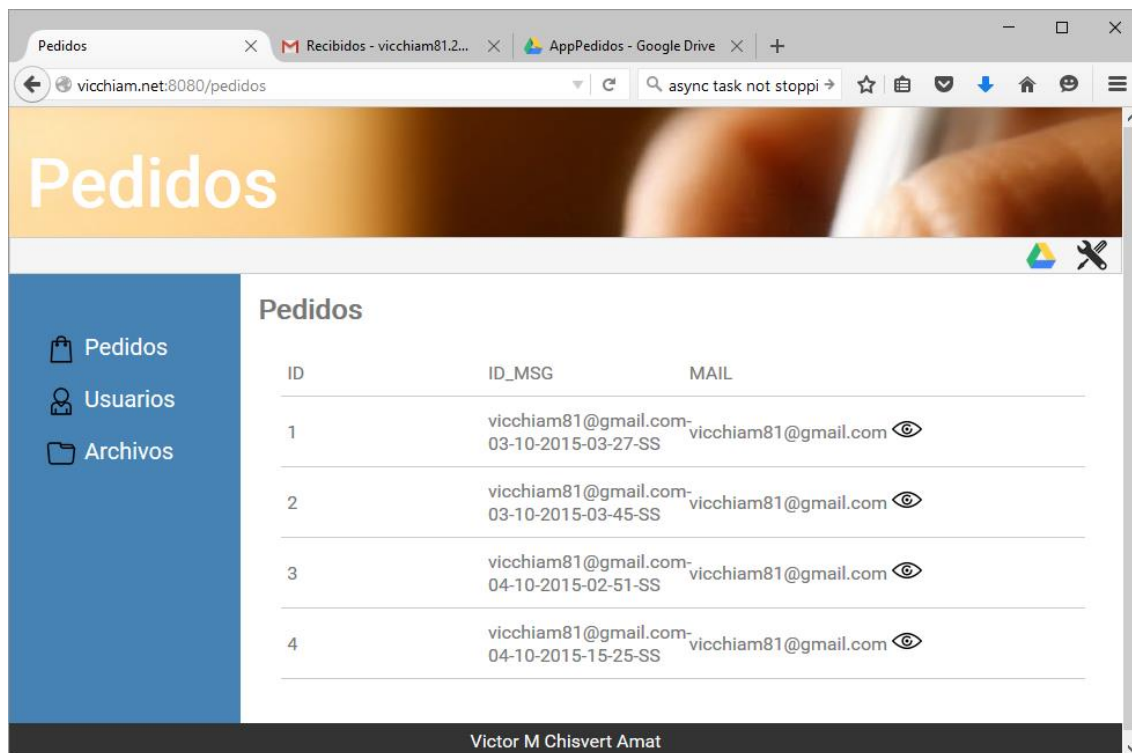
### Sincronizar archivos



Cada vez que se acceda el sistema comprobará si ha habido modificaciones en los archivos, si hay alguna modificación lo indicará mediante el color de los puntos, verde, archivo para agregar a Drive, naranja, archivo para modificar en Drive y rojo archivo para eliminar de Drive.

Al pulsar sobre el icono sincronizar, se empezara la sincronización con la carpeta Drive.

## Pedidos



ID	ID_MSG	MAIL
1	vicchiam81@gmail.com-03-10-2015-03-27-SS	vicchiam81@gmail.com
2	vicchiam81@gmail.com-03-10-2015-03-45-SS	vicchiam81@gmail.com
3	vicchiam81@gmail.com-04-10-2015-02-51-SS	vicchiam81@gmail.com
4	vicchiam81@gmail.com-04-10-2015-15-25-SS	vicchiam81@gmail.com

Cuando se acceda a esta página el sistema comprobará si hay nuevos correos, si es así los descargará y agregará en la base de datos, luego procederá a mostrar todos los pedidos enviados.



## Detalle Pedido

Pedido

Cliente: Stephen Kennerly Archts Inc Pc  
Contacto: Mirta Mallett  
Mail: mirta\_mallett@gmail.com  
Tel: 212-870-1286  
Direccion: 7 S San Marcos Rd New York

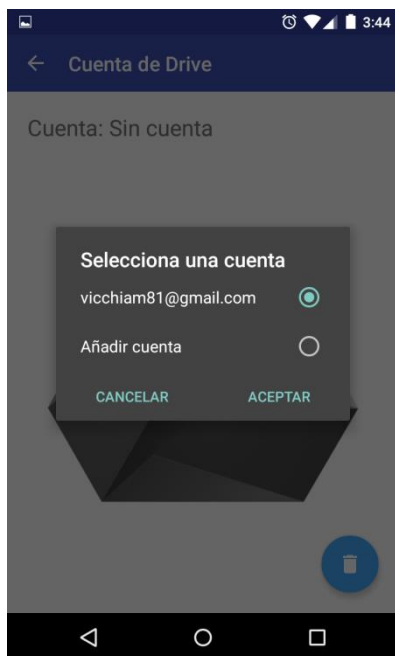
Producto	Precio Und	Cantidad	Desc	Iva	Total
Microsoft Data Analyzer 2002	179	2 und	5%	21%	412 Euros

Victor M Chisvert Amat

Se visualiza el detalle de un pedido enviado por un cliente.

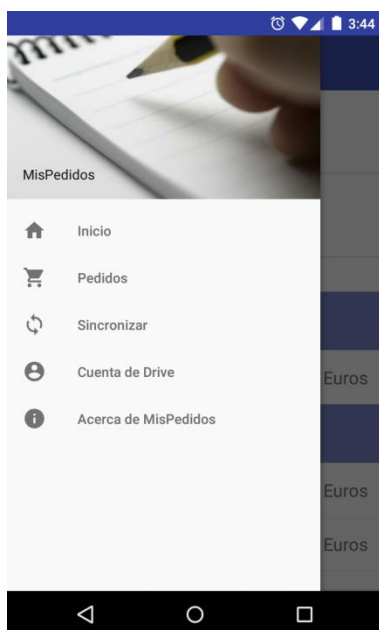
Manual de cliente:

### Inicio de la aplicación



Al iniciare la aplicación por primera vez, esta pedirá una cuenta para conectarse a Drive.

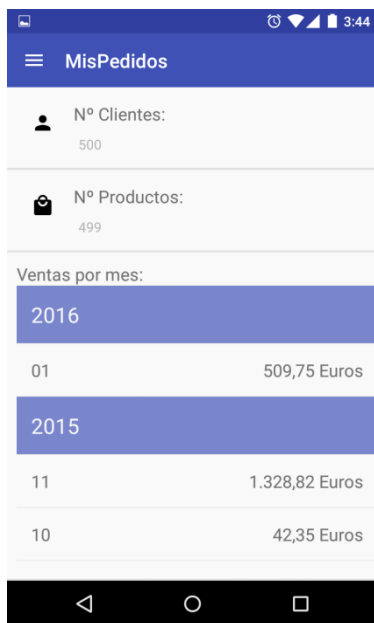
### Página Inicial (Menú)



En el menú principal tenderemos las siguientes opciones:

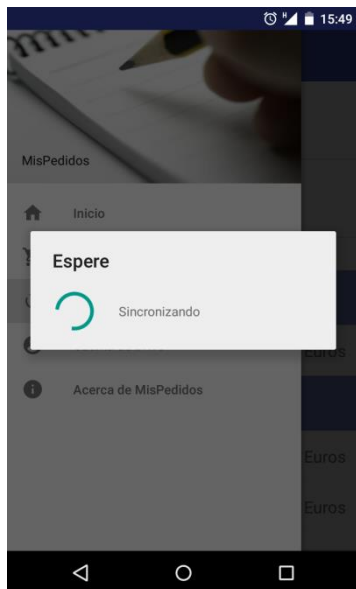
- Inicio, que esconderá el menú, y mostrar la interfaz inicial.
- Pedidos, que mostrar la interfaz de gestión de pedidos.
- Sincronizar, que iniciará la sincronización de datos de Drive con la base de datos del cliente.
- Cuenta de Drive, que iniciará una interfaz que indicará si hay acceso a la carpeta de Drive compartida, y donde se podrá modificar la cuenta de acceso.
- Acerca de..., muestra información sobre la aplicación.

### Interfaz inicial (Menú oculto)



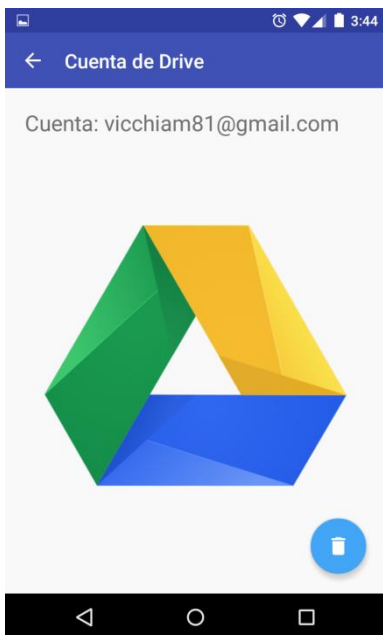
Muestra un resumen de los datos, el número total de clientes y productos en la Base de datos, un resumen agrupado por año y mes del importe total de los pedidos .

### Sincronizar



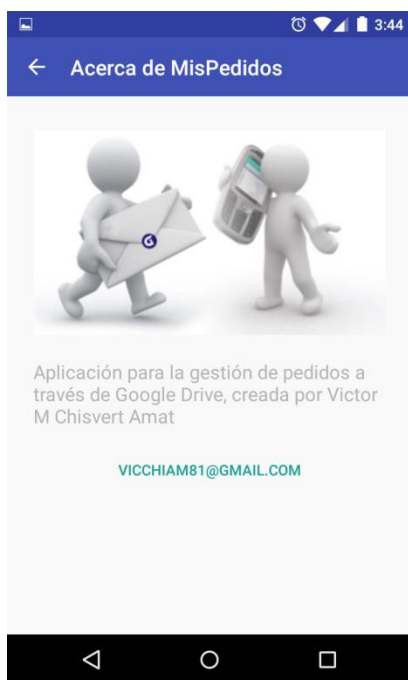
Mensaje durante la sincronización de datos.

### Cuenta de Drive



Comprueba el acceso a Drive y a la carpeta compartida de la aplicación, permite modificar la cuenta de acceso.

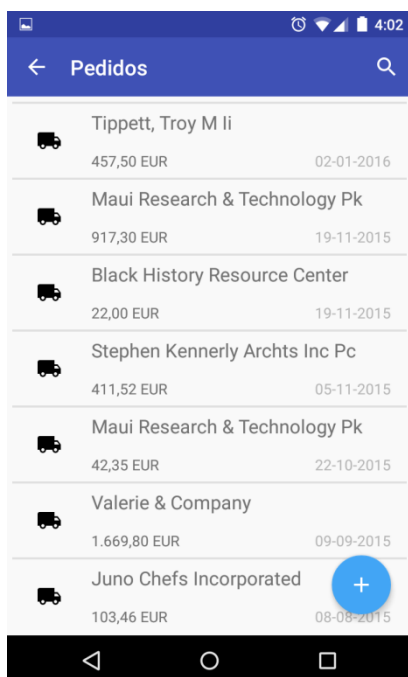
### Información



Muestra información de la aplicación.

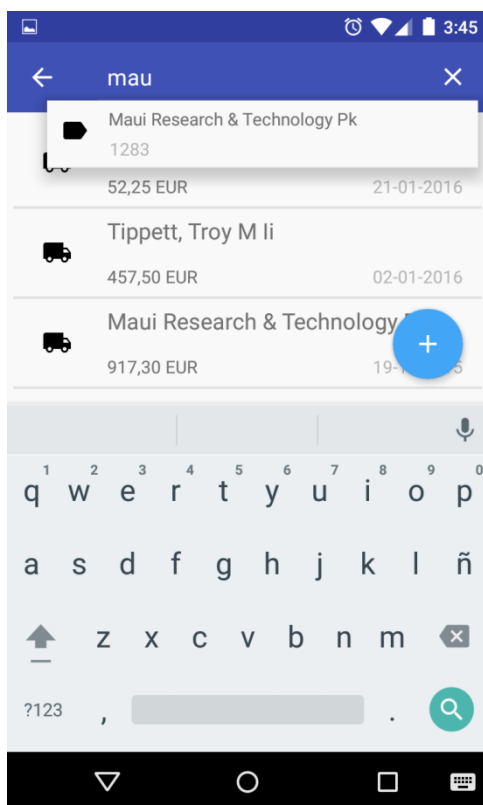


## Pedidos



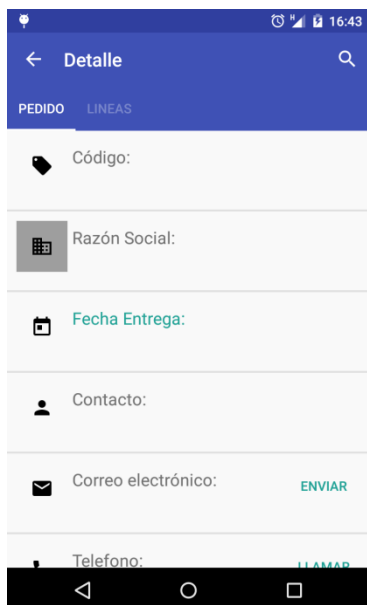
Muestra todos los pedidos de los últimos 6 meses que hay en la base de datos.

## Pedidos (Buscar)



Permite la búsqueda de un pedido por el nombre de la empresa del cliente, búsqueda con autocompletado.

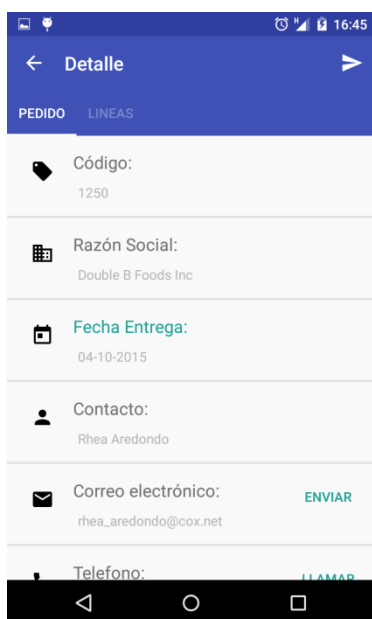
## Agregar/Modificar/Borrar Pedido



Si es un nuevo pedido tiene la opción de buscar un cliente para agregarlo. El cliente se puede buscar por código, razón social, contacto, teléfono, o email, el campo buscar se indica pulsando el icono del campo, este se marcará con fondo gris, al elegir un cliente del listado del autocompletado, se rellenarán los campos.

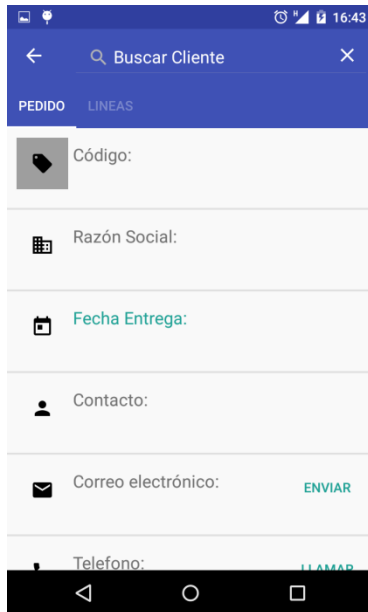
El único campo que se indica por parte del usuario es la Fecha de entrega, que se elegirá mediante un calendario.

Se pueden enviar correos desde el campo Correo electrónico, o llamar desde el campo Teléfono.

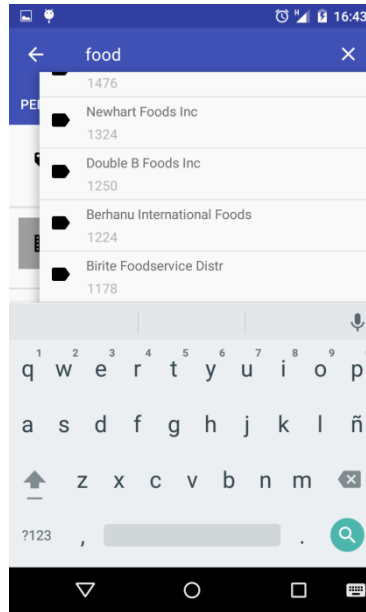


Si el pedido ya está creado no tendrá disponible la utilidad de buscar ya que no se puede modificar el cliente, sí que podrá cambiar la fecha de entrega, siempre que no sea inferior a hoy. También tendrá disponible la opción de enviar, que procederá a abrir el correo electrónico para enviar un adjunto con el pedido a la dirección indicada en el servidor.

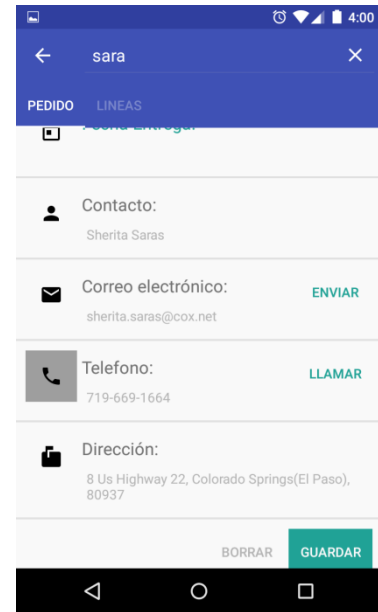
### Cliente Nuevo



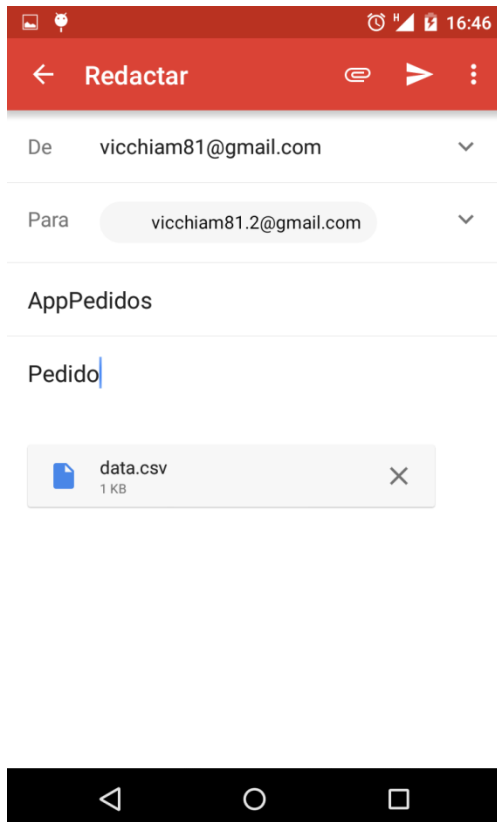
### Búsqueda



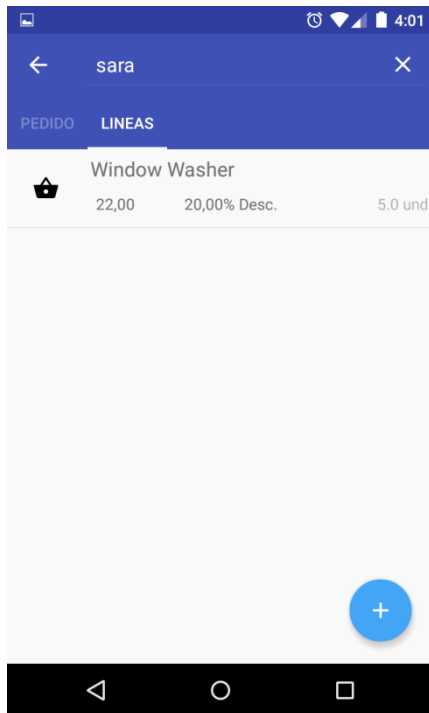
### Cliente agregado



### Envío del correo



## Líneas del pedido



En este interfaz aparecerán las líneas del pedido.

## Agregar/Modificar/Eliminar

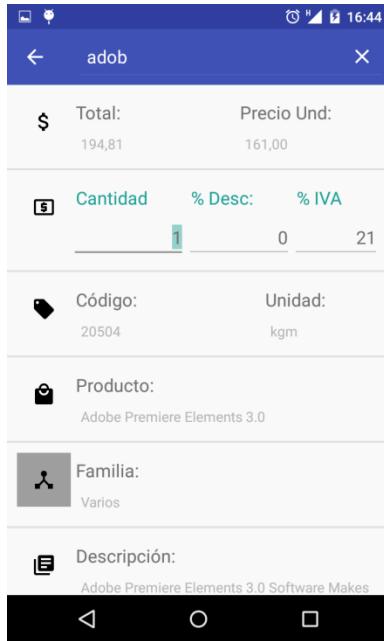


Si es una nueva línea tiene la opción de buscar el producto desde el buscador con autocompletado, al igual que en el la cabecera del pedido se puede buscar por código, producto, y familia.

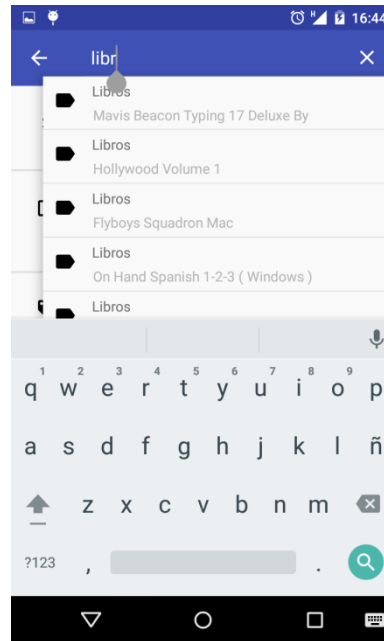
Los campos que puede modificar el usuario son cantidad, el descuento y el IVA.

Si la línea ya existe solo se permitirá modificar los campos editables por el usuario, sin opción de buscar.

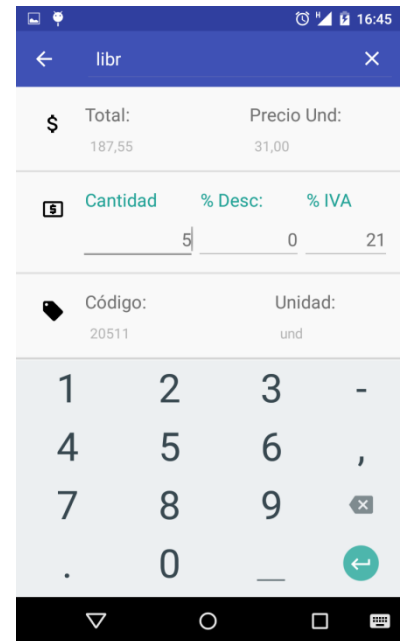
### Selección campo a buscar



### Búsqueda



### Modificar cantidad



Tanto la cabecera del pedido como la línea tienen un botón de borrar si no es nuevo.

