

EASYCOUNT

Victor Quero Chavarría

CFGS (DESARROLLO DE APLICACIONES MULTIPLATAFORMA) Anahí Mula de la Banda

Abstract

El tema principal de este proyecto es la gestión del almacenamiento de productos dentro de una pequeña empresa. Aunque esto pueda resultar un tema muy fácil de abordar el control del inventario es un elemento muy importante para el desarrollo, tanto en grandes empresas como en pequeñas y medianas, además de un factor fundamental para el control de costes y rentabilidad de cualquier hotel o establecimiento turístico.

Por ello ofrecemos el siguiente proyecto para abordar la gestión de dichos inventarios. Durante este documento abordaremos la creación de una aplicación práctica e intuitiva, el análisis de requisitos y el esfuerzo empleado en su desarrollo para convertirse en una aplicación amena a los ojos y conocimientos de los usuarios.

El proyecto usará el principal entorno de desarrollo de Android para su puesta en marcha, donde se codificará el entramado de la aplicación en un lenguaje multiplataforma como lo es Java. Del nombre del proyecto "EsayCount" se especula el especial esfuerzo que se empleara en desarrollar el proyecto para contener un uso sencillo y eficaz.

The main theme of this project is product storage management within a small business. Although this may be a very easy subject to address, inventory control is a very important element for development, both in large companies and in small and medium-sized ones, as well as being a fundamental factor for cost control and profitability of any hotel or tourist establishment.

For this reason, we offer the next project for the management of such inventories. During this document we will address the creation of a practical and intuitive application, the analysis of requirements and the effort spent in its development to become a pleasant application in the eyes and knowledge of users.

The project will use the main Android development environment for his implementation, where the framework of the application will be encoded in a multiplatform language such as Java. From the name of the project "EsayCount" it is speculated the special effort that was used in developing the project to contain a simple and effective use.

Contenido

Introducción	4
Descripción del Proyecto.....	4
Objetivos del Proyecto	4
Análisis.....	5
Casos de uso	5
Diagrama de clases.....	5
DIAGRAMA DE GRANTT.....	6
Análisis de Requisitos	7
Contexto Empresarial.....	9
Desarrollo	11
Diseño.....	11
Código principal.....	22
Bases de Datos y clases adicionales	26
Herramientas usadas y futuras actualizaciones	27
Conclusiones	29
Bibliografía	30
Anexos.....	31
Manual de Usuario	31
La importancia de los inventarios	36
La tendencia el desarrollo en Android	37

Introducción

Descripción del Proyecto

Este proyecto nace de un estudio realizado a varias empresas dedicadas al sector alimenticio en el cual se demostró un amplio interés por contar con algún tipo de software móvil que aligerara la carga de establecer y gestionar los inventarios.

El proyecto consiste en una aplicación Android capaz de gestionar el inventario de una pequeña empresa de alimentación en la cual el administrador de la aplicación podrá gestionar una base local en SQLite, eliminando, modificando o consultando artículos mediante un código identificativo. Asimismo, dispondrá de una base de datos con proveedores mediante la cual se podrá establecer contacto telefónico con ellos.

Objetivos del Proyecto

Los objetivos de este proyecto son los siguientes:

- Establecer un sistema de registro de productos sencillo y funcional apto para pequeños negocios alimenticios
- Contar con dos bases de datos para almacenar tanto los productos como los proveedores
- Permitir el uso de la base de datos para que el cliente pueda gestionar el inventario de su negocio
- Elaborar una aplicación práctica que permita contactar con un proveedor al verificar la escasez de un producto

Análisis

Casos de uso

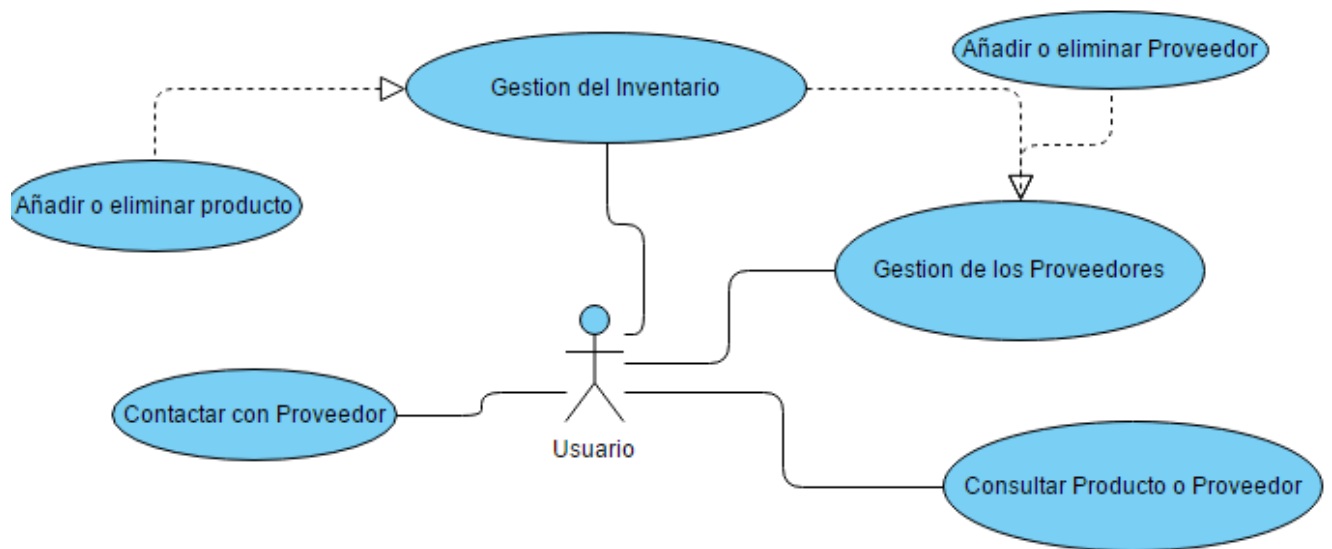


Diagrama de clases

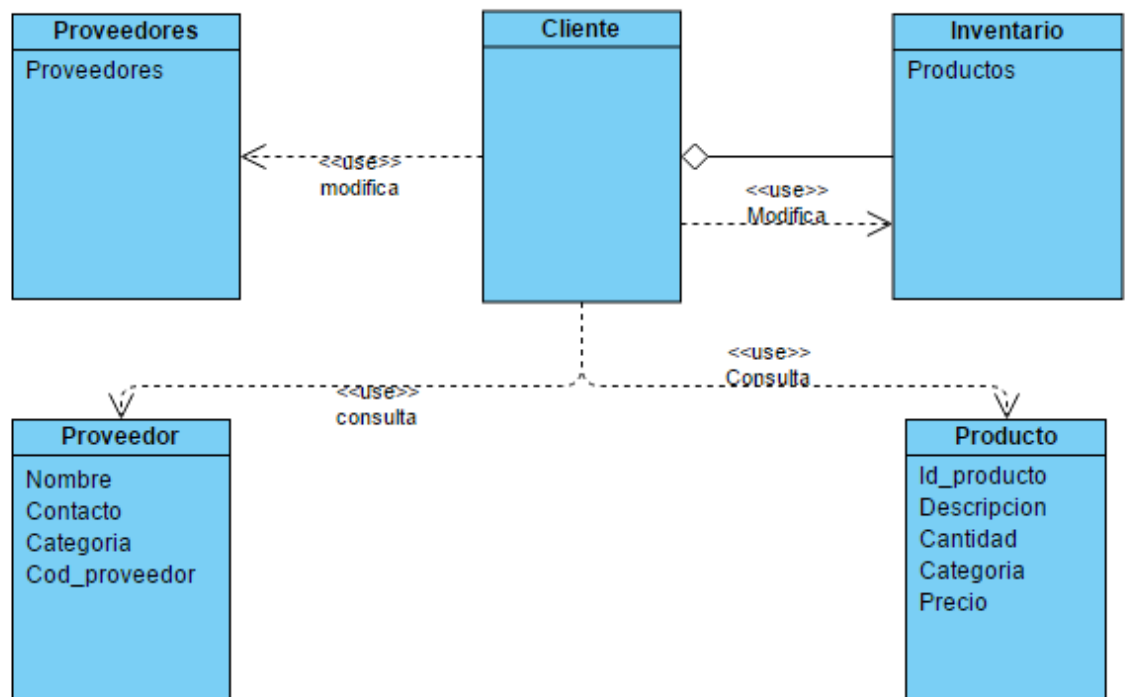
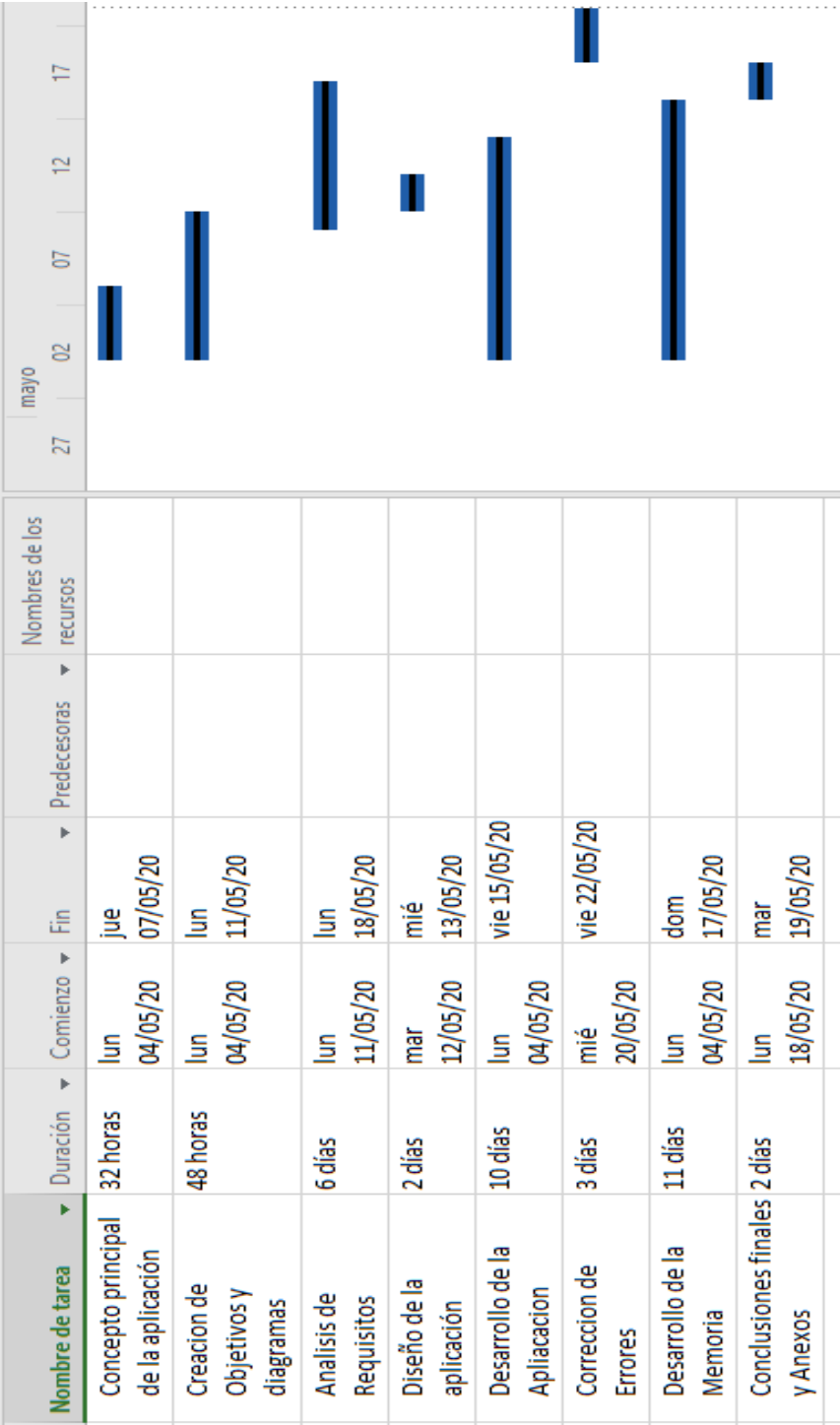


DIAGRAMA DE GRANTT



Análisis de Requisitos

El control de inventario es un elemento importantísimo en el desarrollo de cualquier empresa, además de un factor fundamental para el control de costes y rentabilidad. Por eso mismo la mayoría de las empresas se ven obligadas a realizar inventarios para determinar su valor a efectos fiscales.

Por consiguiente, tener una buena herramienta que mitigue el stress para los equipos humanos empresariales es igual de importante que implementar una sistemática adecuada. En definitiva, lo que se pretende con este proyecto es aportar una solución clara, robusta y económica para los inventarios. Con eso en mente se elaboró un análisis de requisitos basado en una encuesta previa realizada a futuros usuarios.

Durante la encuesta se abordaron diversos y como resultante dio lugar al consiguiente proyecto. Dentro de dicho proyecto la aplicación que se realizara varias funciones como son la gestión de dos bases de datos y la realización de llamadas.

Para la elaboración de la aplicación se requerirá de un software específico con el cual mejorar su gestión y establecer un entorno sencillo y accesible al usuario. Gracias al uso de un entorno de desarrollo como Android Studio y sus módulos tanto de bibliotecas como de motor de aplicaciones para Google. Los cuales nos permitirán mejorar la calidad del código y aumentar la productividad de los desarrolladores gracias a su función de análisis de programación. Asimismo, mediante los resultados de la encuesta anteriormente mencionada podemos desglosar este análisis de requisitos en tres partes:

- Requisitos en cuanto a diseño: Las opiniones de los encuestados fueron bastante dispares sobre el diseño que buscaban en la aplicación. Sin embargo, muchos de ellos coincidieron en la necesidad de un diseño no muy complejo y que fuera visualmente

ameno y entendible. De acuerdo a estos resultados se intentará establecer una interfaz gráfica practica y funcional donde se equilibre el diseño artístico y el diseño practico.

- **Requisitos prácticos:** Una de las cuestiones que se propuso a los encuestados fue que uso le darían a una aplicación de stock que ellos crearan. Y tras comparas las respuestas se llegó a un punto común: la gestión de múltiple de inventario y proveedores. Además, se decidió establecer una función de llama para mayor eficiencia útil dentro de su ámbito operacional. Por ello surgió la necesidad de establecer básicamente una app que manejara ambos conceptos.
- **Requisitos de hardware y software:** Durante la encuesta se mostró que la mayoría de los futuros clientes parecían más dispuestos a interactuar con una app móvil que con un programa en ordenador, esto llevo a plantearnos el hacer uso del IDE de Android Studio que mencionamos antes. Así pues, el hardware necesario o requerido para la ejecución de la aplicación se cualquier dispositivo compatible con el sistema operativo Android (el más común entre los encuestados) o en su defecto un dispositivo o interfaz que permita realizar su emulación como podría ser la herramienta BlueStacks entre otras. En cuanto software los requisitos mínimos aparte del ya antes nombrado sistema operativo Android o compatibles serian la disposición de un espacio de memoria amplio y la concesión de permisos para contactos para la función de llamada que se integrara en la aplicación.
- **Requisitos de seguridad:** La seguridad de la aplicación se verá implementada mediante el ERP de Odoo y su módulo de gestión de incidencias. A través del cual se establecería contacto con el cliente y se dispondrían los medios para la solución del problema acontecido.

Contexto Empresarial

Este proyecto está diseñado y creado originalmente por una empresa líder en el sector de las aplicaciones móviles. Con más de 25 años de experiencia y repleta de casos de éxito en la implementación de aplicaciones y herramientas informáticas al servicio de empresas y particulares debido a su filosofía empresarial y a la alta cualificación de sus desarrolladores.

La empresa cuenta con un gran número de trabajadores capacitados y motivados para ofrecer soluciones integrales con la máxima calidad, los cuales reciben una constante formación. Dentro de sus numerosos centros de desarrollo a lo largo del mundo ha presentado numerosas apps valoradas positivamente por la gran mayoría de los usuarios de las mismas.

Para la realización de este proyecto se estableció un grupo de 5 trabajadores entre los cuales se estableció un sistema organizativo basado en nuestra filosofía empresarial. Mediante el cual se consiguió reducir costes y mejorar la eficiencia de dichos trabajadores de tal manera que se logró crear una aplicación funcional en un corto periodo de tiempo.

En cuanto a costes el presupuesto variara debido a varios factores entre ellos cabe destacar:

- La funcionalidad
- La interfaz de usuario y el diseño
- La plataforma de lanzamiento

En definitiva, tras una investigación de costes especulamos que el proyecto actual supondrá un valor de 41,25 €/hora en función del tiempo a desarrollarlo. Esta cifra no tiene por qué ser el coste final del proyecto puesto que se excluyó el salario de los trabajadores, así como posibles imprevistos que pudieran originarse durante el desarrollo del proyecto.

Son muchos los factores que nos facultan para conseguir este objetivo, entre los que podemos destacar la utilización de las tecnologías más avanzadas, la aportación de las personas más idóneas para cada proyecto y el alto nivel de cualificación de los técnicos y colaboradores de nuestra empresa. El conocimiento de nuestros técnicos abarca los principales entornos tecnológicos utilizados en el mercado: Bases de datos (DB2, Oracle, SQL Server, MySQL), Lenguajes de Programación (J2EE, JAVA, COBOL, NATURAL, .NET),

Aplicaciones de gestión empresarial (SAP, BI...).

Nuestro portfolio de actividades abarca desde:

- Mantenimiento y soporte tecnológico
- Desarrollos a medida de productos y componentes
- Gestión de aplicaciones y desarrollo del software
- Análisis y Diseño de soluciones
- Optimización y calidad de aplicaciones
- Outsourcing tecnológico

Además, desde nuestra empresa velamos por la información confidencial y de carácter personal de todos los datos de nuestros trabajadores y en cumplimiento con la Ley Orgánica de Protección de datos, en el 2011 hemos implantado todo el procedimiento de seguridad que asegura este derecho.

Los *softwares* que creamos son modulares y permiten su crecimiento, adaptación y cambios a medio-largo plazo. Esta vía de ejecución ofrece un claro ahorro de costes a los clientes, solución de sus necesidades en cortos períodos de tiempo y la posibilidad de adaptarnos a nuevas necesidades en tiempo real. Por todo esto nuestra empresa se ha coronado como primordial en su sector y es recomendada ampliamente por cientos de clientes.

Desarrollo

Ahora procederé a desmenuzar el proyecto en tres partes

Diseño

Para empezar, se usará una pantalla en colores fríos que incluirá el logo de la compañía y el nombre del desarrollador principal.



Ilustración 1 Portada de la aplicación

A continuación, describiré la pantalla base de la aplicación a desarrollar, la cual se compondrá básicamente de cuatro botones con los cuales se podrá interactuar con el inventario y un menú en la parte superior derecha con el que poder acceder a otras pantallas adyacentes.



2 Pantalla Base y menú superior

Como puede observarse en la imagen los botones serán iconos propios del de entorno de desarrollo para evitar costes. Una vez pulsemos en cualquiera de ellos nos abrirán otra pantalla:

- 1- Opción Añadir: En esta pantalla encontraremos varios campos Edit Text donde rellenar la información del producto a catalogar



EasyCount

Rellene los campos para formular la inserccion

ID

Producto

Cantidad

Categoria

Precio

CONFIRMAR

ATRAS

3 Pantalla de inserción

- 2- Opción Consultar: En esta pantalla contaremos con un único Edit Text en el cual se introducirá el campo id del producto a busca y posteriormente aparecerá debajo del botón en su respectivo TextView

Introduzca el ID del producto

BUSCAR

4 Pantalla de Consulta

- 3- Opción Editar: Mostrara una serie de campos a rellenar por el usuario de forma que modifique lo existente

Rellene para modificar el producto

ID
Producto
Cantidad
Categoria
Precio

CONFIRMAR

ATRÁS

5 Pantalla de Modificación

- 4- Opción Eliminar: Mostrará a un campo a rellenar donde se ingresará el id del producto a suprimir

Introduzca el codigo del
producto a eliminar

Codigo de producto

CONFIRMAR

6 Pantalla Borrado

En la pantalla de proveedores nos encontraremos un listview con los distintos proveedores, en el cual al pulsar sobre ellos podremos llamarlos, así como dos botones uno para agregar más y otro para borrado.



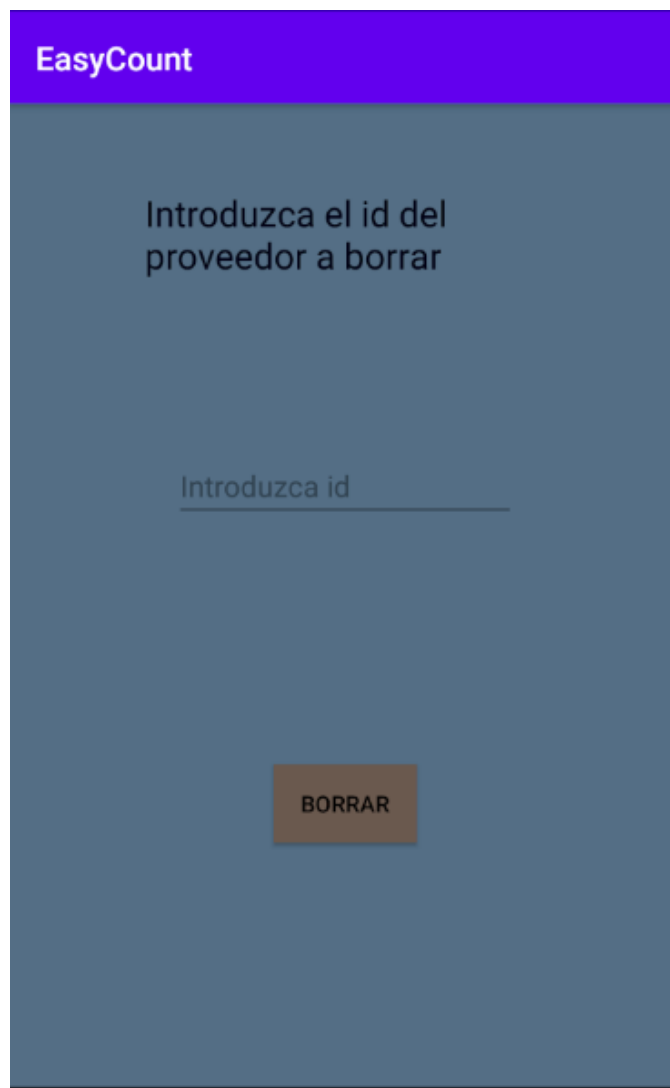
7 Listview Proveedores

Al pulsar el botón de Añadir abriremos otra pantalla en la cual rellenaremos los datos del proveedor y mediante un radiogroup seleccionaremos la categoría de este.

The screenshot shows the 'EasyCount' app interface. At the top is a purple header with the text 'EasyCount'. Below the header, the main area has a dark blue background. The text 'Rellene los campos' (Fill in the fields) is centered. There are two input fields: 'Introduzca el nombre' (Enter the name) and 'Introduzca el telefono' (Enter the phone number). Below these fields are three radio buttons with labels: 'Congelados' (Frozen), 'Alimentacion' (Feeding), and 'Bolleria' (Bakery). At the bottom, there are two buttons: 'ATRAS' (Back) and 'CONFIRMAR' (Confirm).

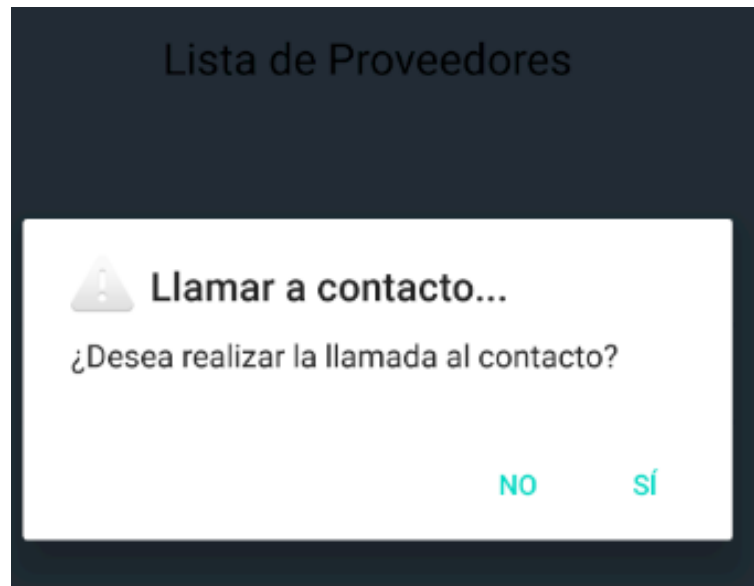
8 Añadir Proveedor

Por otro lado, al pulsar en el botón de borrado nos iremos a una pantalla simple donde introduciendo el id del proveedor que será visible en la pantalla principal podremos borrarlo satisfactoriamente.



9 Pantalla de Borrado Proveedores

Volviendo a la pantalla con la Listview queda resaltar la capacidad de pulsar en los miembros listados para llamar directamente al contacto. Esto hará emerger una pequeña ventana de color blanco en la cual podremos confirmar o no la llamada.



10 Pantalla de proveedores

Y por último en una única pantalla se mostrará un breve agradecimiento al usuario de la aplicación y un TextView con la versión de la misma

Bienvenido a la pantalla informacion

Esta aplicacion fue diseñada por
la compañía TFG para el manejo
del stock en pequeñas empresas.
Esperamos que sea de su agrado.



Version 1.0

11 Pantalla de Agradecimiento

Código principal

En cuanto a uso de código cabe destacar el uso de diversas clases específicas de Java y SQLite. En primer lugar, para la primera pantalla de la aplicación usaremos un handler temporal que tras unos segundos nos enviara directo a la pantalla principal. Estos handlers nos permitirán comunicarnos desde un subproceso con el hilo principal. Una vez en la pantalla principal procederemos primeramente a enlazar las constantes con sus respectivos iconos cliqueables. Para posteriormente crear varios métodos que abrirán distintas pantallas donde el usuario gestionara la primera de las dos bases de datos coexistentes. Asimismo, usando la herramienta XML para menús de Android Studio implementaremos un menú en la parte superior derecha de nuestra app con tres ítems que nos redirigirán a las pantallas de inventario, proveedores y agradecimientos respectivamente.

Una vez pulsados los diversos botones de la pantalla principal nos redirigirán a varias pantallas donde podremos realizar tales operaciones como son la inserción, la modificación, el borrado y la consulta de la base de datos inventario. Todas esto posible gracias a la implementación de la clase SQLiteOpenHelper usada en la creación de la base de datos.

```
public BaseDeDatosInventario(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table inventario(id int primary key, producto varchar(30),cantidad int,categoria varchar(30),precio double)");
}
```

A continuación, hablaremos del contenido de las pantallas antes mencionadas:

- Pantalla Añadir: Estará formada por varias entradas de texto con las que al pulsar un botón y usando la clase *android.content.ContentValues* con la cual adjuntar los valores de las entradas de texto a un registro en la tabla inventario.
- Pantalla Modificar: El contenido de su código será similar al de la pantalla anterior, pero incluirá el método *update* de la clase *SQLiteDatabase* pasando el nombre de la tabla, el objeto de la clase *ContentValues* y una condición establecida en una cláusula *where* (en nuestro caso el contenido de las entradas de texto)

```
bd.update( table: "inventario", contenido, whereClause: "id="+valor_id, whereArgs: null);
```

- Pantalla Borrar: Contendrá un campo de introducción de texto y un botón que al pulsarlo ejecutara una sentencia condicional en la cual si el campo ha sido rellenado con el valor de un id, borrara ese registro de la base de datos. Para ello usara el método delete de la clase SQLiteDatabase.

```
bd.delete( table: "inventario", whereClause: "id="+id, whereArgs: null);
```

- Pantalla Consulta: Estará formada por un campo donde introduciremos texto y un botón cuya función es mostrar en un TextView el resultado de la consulta. Esto se conseguirá mediante el empleo del método.rawQuery que nos permitirá colocar una sentencia SQL con la cual acceder al registro deseado en la base de datos.

```
rawQuery( sql: "select producto, cantidad , categoria from inventario where id="+id_producto, selectionArgs: null);
```

Eso era todo en cuanto a la pantalla principal, ahora pasaremos a hablar de la pantalla Proveedores la cual contará con un listview cuyo principal contenido será una consulta a la base de datos de proveedores. Para ello se usarán dos métodos Consultar y SacarLista en los cuales consultaremos la tabla proveedor y mostraremos su resultado a través de un ArrayList (clase que permite almacenar datos en memoria) respectivamente. Una vez sacada la lista a consultar emplearemos el método SetOnItemClickListener, para poder pulsar en ella, para posteriormente generar un dialogo de alerta que nos cuestione si deseamos o no proseguir a la llamada del contacto seleccionado.

```
lista.setOnItemClickListener((parent, view, position, id) → {  
    AlertDialog.Builder alertDialog = new AlertDialog.Builder( context: Proveedores.this);  
    alertDialog.setMessage("¿Desea realizar la llamada al contacto?");  
    alertDialog.setTitle("Llamar a contacto...");  
    alertDialog.setIcon(android.R.drawable.ic_dialog_alert);  
    alertDialog.setCancelable(false);
```


Así una vez confirmada la operación se ejecute la clase Toast mostrando el teléfono del proveedor solicitado. Asimismo, la pantalla Proveedores contendrá dos botones para añadir o eliminar contenido de la base de datos. Ambos redirigieran al usuario a otras pantallas, siendo la primera de ellas la pantalla Nuevo Proveedor en la cual habrá una serie de Edit Text y un radiogroup donde se especificarán los detalles del proveedor. El código que se empleara será similar a cualquier otro agregado a la base de datos salvo por el hecho de que se usara un condicional para especificar cuál fue el radiobutton seleccionado.

```
if(con.isChecked()){  
    cate="Congelados";  
}  
if(ali.isChecked()){  
    cate="Alimentos";  
}  
if(bol.isChecked()){  
    cate="Bolleria";  
}
```

Por último, para el desarrollo de la pantalla Borrar_proveedor se volverá a emplear el método de la clase SQLiteDatabase anteriormente citado para poder suprimir el registro deseado de la tabla proveedor.

```
SQLiteDatabase db=admin.getReadableDatabase();  
String id=ed.getText().toString();  
if(!id.isEmpty())  
{  
    int cantidad=db.delete(utilidades.TABLA_proveedor, whereClause: utilidades.CAMPO_ID+"="+id, whereArgs: null);  
    Toast.makeText(context, this, text: "He borrado "+cantidad, Toast.LENGTH_LONG).show();  
}
```

Además, tanto la pantalla de Nuevo Proveedor como Borrar-proveedor dispondrán de un handler temporal que los devolverá automáticamente a la pantalla Proveedores. Hasta aquí el código principal del programa ahora hablaremos de las bases de datos y las clases adicionales usadas.

Bases de Datos y clases adicionales

Para nuestro proyecto empleamos SQLite, el cual es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL autocontenido, sin servidor y sin configuración. La primera base de datos que crearemos será la de inventario mediante el método onCreate de SQLiteHelper. Este método es bastante sencillo ya que solo necesitaremos extender la clase SQLiteHelper y de ahí crear la base de datos.

Mientras que para la base de datos de Proveedores necesitaremos la creación previa de dos clases. La clase proveedor que contendrá los atributos de la tabla y la clase utilidades que enlazará ambas.

```
public class utilidades
{
    public static final String TABLA_proveedor="proveedor";
    public static final String CAMPO_ID="id";
    public static final String CAMPO_NOMBRE="nombre";
    public static final String CAMPO_TELEF="telefono";
    public static final String CAMPO_CATEG="categoria";

    public static final String Crear_Tabla_Proveedor="Create table "+TABLA_proveedor+"("+CAMPO_ID+" Integer,"+CAMPO_NOMBRE+" text, "+CAMPO_TELEF+" text,"+CAMPO_CATEG+" text)";
```

Finalmente creadas ambas clases procedemos a crear la base de datos de los proveedores

```
public class BDProveedores extends SQLiteOpenHelper
{
    public BDProveedores(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version); }

    @Override
    public void onCreate(SQLiteDatabase db) { db.execSQL(utilidades.Crear_Tabla_Proveedor); }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("Drop Table if exists proveedor ");
        onCreate(db);
    }
}
```

Herramientas usadas y futuras actualizaciones

Las herramientas usadas para este proyecto serán:

- El IDE oficial de desarrollo de aplicaciones Android basado en IntelliJ, llamado Android Studio. Mediante la cual se creará la aplicación en lenguaje java.
- EL Sistema ERP de Odoo como medio para gestionar cualquier contratiempo derivado del uso de la aplicación por los clientes, además de establecer la seguridad para la app mediante el login en el mismo.
- El lenguaje de programación de Java para Android Studio debido a su virtud de multiplataforma, la facilidad en su manejo y otras múltiples ventajas que ofrece dentro del desarrollo de una aplicación móvil
- La base de datos de SQLite gracias a la no dependencia externa, la gran cantidad de librerías compatibles con distintos lenguajes de programación y la capacidad de soportar diversas gestiones en las tablas

La aplicación a desarrollar nace del pensamiento de sencillez de ahí que las bases de datos no estén relacionadas directamente, sin embargo, esto podría cambiar en futuras actualizaciones. En las cuales entre otros podrían implementarse el uso de una base de datos externa, la gestión de usuarios mediante registro o sistemas de alarma ante la escasez de cierto producto. Dichas actualizaciones contarían con cambios tanto a nivel de diseño como de programación. Asimismo, se crearía una página web oficial de la aplicación para establecer un contacto más directo con los usuarios.

Por otro lado, Android es un sistema operativo que se actualiza continuamente, por lo que debería programarse una tarea de mantenimiento para estudiar nuevas versiones y adaptar el proyecto, evitando de este modo comportamientos erróneos de la aplicación en futuras versiones de Android.

Conclusiones

El desarrollo de esta aplicación fue bastante laborioso pero gratificante. Desde un principio fue bastante difícil escoger una idea a desarrollar como proyecto. Para empezar durante este proyecto no conté con el apoyo formativo de unas prácticas en empresa con las cuales mejorar el rendimiento organizativo y de desarrollo de la aplicación. Hubo muchas preguntas, dudas de todo tipo, afortunadamente hoy día a través de Internet tenemos acceso a un gran volumen de información, pero que hay que cribar, seleccionar y asimilar, tampoco es tan fácil no está todo lo que se necesita en cualquier momento, hay que recurrir también a libros o intercambio de conversaciones con otras personas.

Atendiendo al desarrollo técnico del proyecto, fue difícil sintetizar todas las ideas y requisitos que había sobre la mesa. Una vez realizadas las fases de análisis y diseño, las fases posteriores fueron encauzadas rápidamente. Por lo que las primeras fases de planificación, análisis y diseño resultaron ser de suma importancia.

En líneas generales creo haber logrado alcanzar los objetivos y requisitos propuestos de manera eficiente y eficaz. Asimismo, creo que de manera general he logrado crear el proyecto que idee. Cabe destacar también que gracias a dicho proyecto he aprendido a organizarme, a buscar información precisa y a desarrollar un proyecto de principio a fin. No sé si este programa tendrá una gran aceptación entre las personas en general, pero yo me siento muy feliz en su creación.

Por último, dar las gracias tanto a mis profesores como a mis tutores sin los cuales no hubiera podido realizar este proyecto, para el cual no contaba con ninguna experiencia previa.

Bibliografía

- 1- <https://www.noray.com/blog/la-importancia-de-realizar-inventarios-en-nuestra-empresa/>
- 2- <https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- 3- <https://www.xatakamovil.com/sistemas-operativos/asi-como-android-se-ha-comido-mercado-diez-anos>
- 4- <https://espai.stucom.com/tecnologia/proyecto-final-de-dam-spring-data-rest/>
- 5- <https://repositorio.comillas.edu/rest/bitstreams/715/retrieve>
- 6- https://repositorio.uam.es/bitstream/handle/10486/662281/gomez_matesanz_alfonso_tfg.pdf?sequence=1
- 7- <https://pickaso.com/2019/desarrollo-apps-diferencias-android-vs-ios>
- 8- <https://ajpdsoft.com/modules.php?name=News&file=print&sid=537>
- 9- <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper>
- 10- <https://es.stackoverflow.com/questions/143104/cargar-datos-de-sqlite-en-listview>
- 11- <http://www.proyectosimio.com/es/programacion-android-ejemplo-de-bases-de-datos-listview-i/>
- 12- <https://developer.android.com/reference/android/os/Handler>
- 13- <https://blog.ida.cl/disenio/elementos-interfaz-calman-usuarios/>
- 14- https://www.odoo.com/documentation/user/9.0/es/project/advanced/claim_issue.html
- 15- <https://www.qubiq.es/blog/manuales-odoo-1/post/gestion-de-las-incidencias-con-odoo-47>
- 16- <https://www.40defiebre.com/elementos-buena-interfaz-usuario>
- 17- <https://academiaandroid.com/sqlite-android-creacion-acceso-base-datos-insercion/>

Anexos

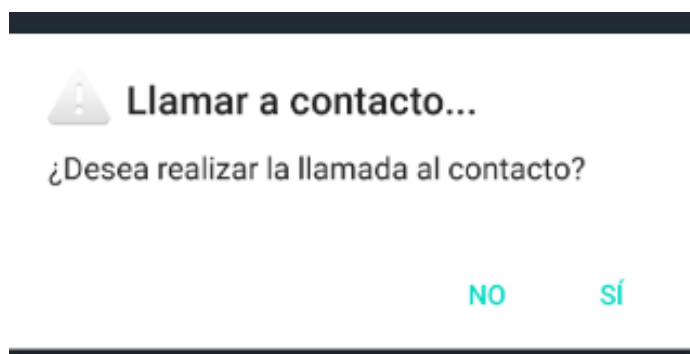
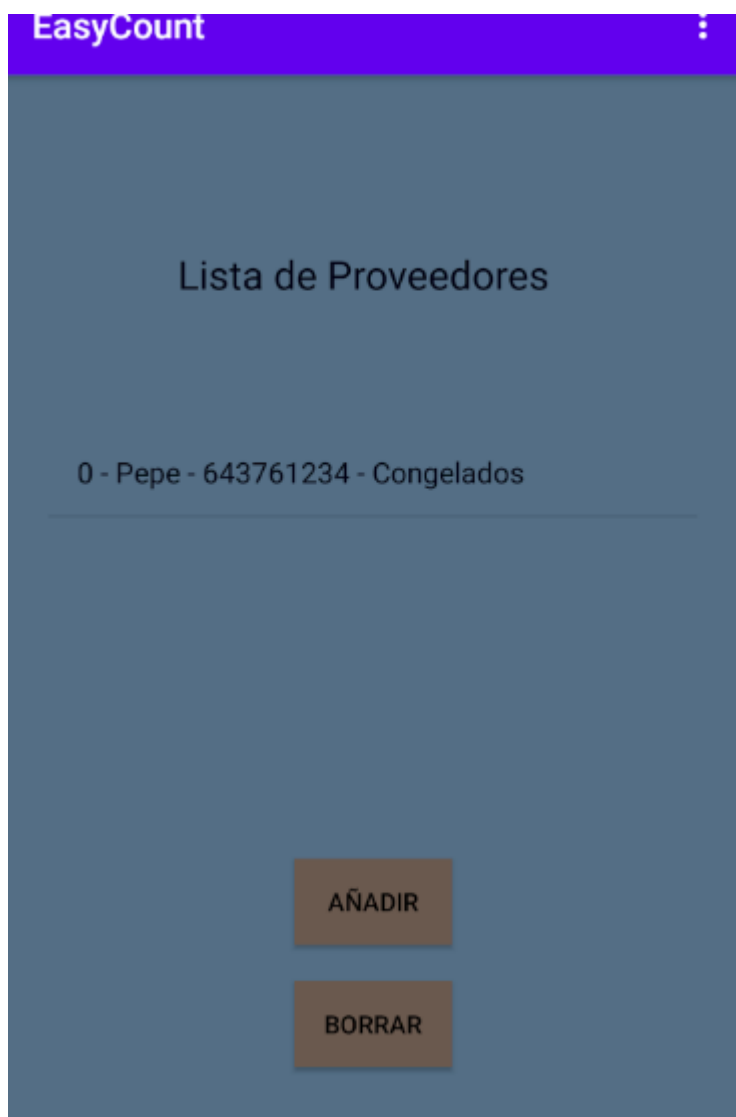
Manual de Usuario

El siguiente manual está diseñado para ser una guía básica para la utilización de la aplicación de dispositivo móvil EasyCount destinada a la gestión del inventario por parte de pequeñas y medianas empresas.

En la página principal de dicha herramienta es posible acceder a un menú desplegable que nos lleve a ver la lista de proveedores o información sobre la aplicación. Asimismo, en esta pantalla también será posible gestionar el inventario mediante los diferentes botones contenidos en la misma.



Al acceder a la lista de proveedores mediante el menú podremos realizar distintas acciones mediante los botones para actualizar la lista pulsando en la lista. Esto último hará que emerja una ventana con la cual podemos efectuar una llamada al proveedor listado.

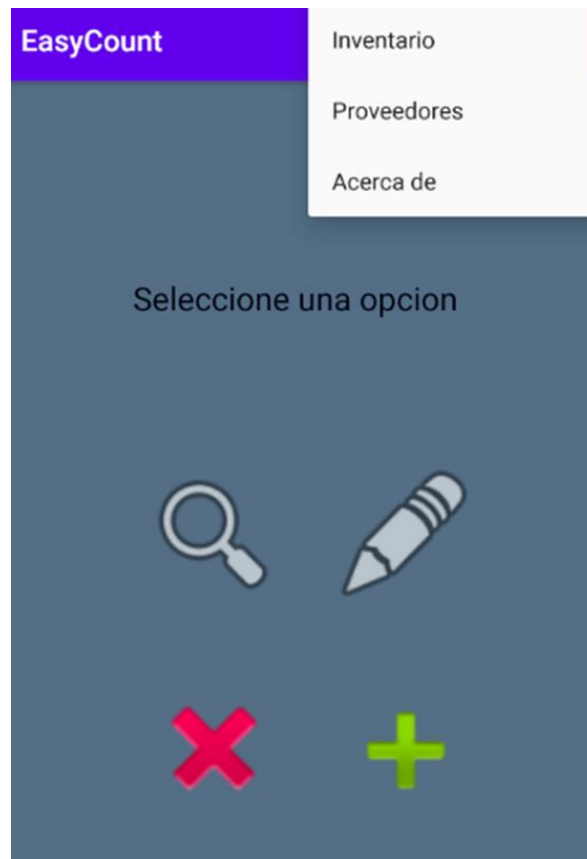


Por último, el usuario puede acceder a una pantalla de agradecimientos que consta de la versión de la aplicación.

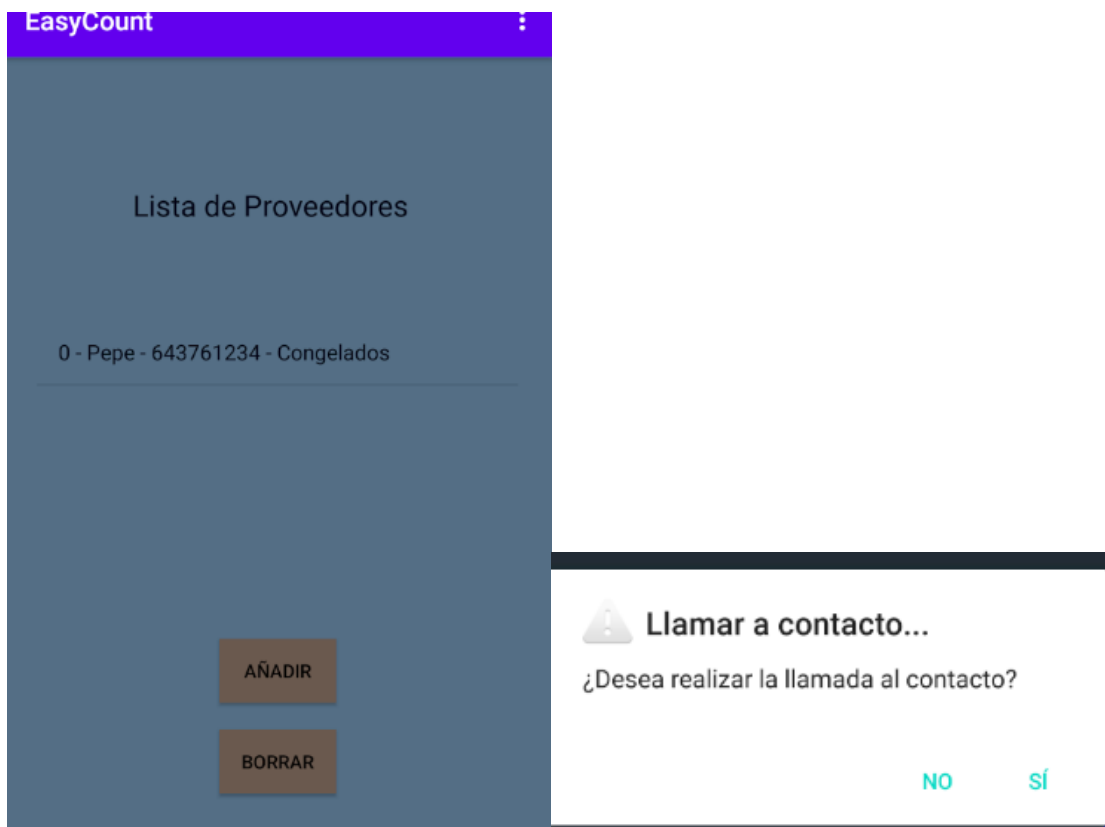


The following manual is designed to be a basic guide for the use of the EsayCount mobile device application for inventory management by small and medium-sized companies.

On the main page of this tool it is possible to access a drop-down menu that takes us to see the list of providers or information about the application. Likewise, on this screen it will also be possible to manage the inventory using the different buttons contained therein.



When accessing the list of suppliers through the menu, we can perform different actions using the buttons to update the list by clicking on the list. The latter will bring up a window with which we can make a call to the listed provider.



Finally, the user can access a thank you screen that consists of the version of the application.



La importancia de los inventarios

La necesidad de inventario dependerá mucho del tipo de empresa y también de la clase de producto con que se trabaje. Si el producto según se recibe, se registra y se guarda en un almacén del que sólo saldrá con una orden de pedido, el inventario puede espaciarse más en el tiempo. Sin embargo, si no existe un registro exhaustivo de los bienes cuando llegan y/o si éstos están al alcance de la clientela, por ejemplo, en estanterías o mostradores, es conveniente aumentar la frecuencia de inventario.

El inventario es, en definitiva:

- Medida de control: para saber lo que se tiene y lo que falta y también para detectar posibles anomalías o problemas, por ejemplo, con proveedores o empleados.
- Herramienta de cálculo y previsión de gasto: que ayuda a conocer en qué se invierte el dinero y reporta mayores ganancias en el negocio.
- Imagen de la empresa: porque si el cliente siempre encuentra lo que quiere y se le ofrece en las mejores condiciones, la imagen que su negocio proyecta es la que debería ser.

Para realizar inventarios lo ideal es automatizar los procesos, así se eliminan subjetividades y se descartan errores humanos, que son las principales causas de desviaciones en los inventarios con respecto a la realidad. Existen en el mercado bastantes opciones a la hora de instalar softwares que lo permitan y cada día las posibilidades crecen gracias a internet. Tener o no stock, o realizar inventarios, depende como hemos dicho de cada empresa, pero hay que tener en cuenta que siempre debe buscarse la máxima eficiencia y rentabilidad para cada tipo de negocio.

La tendencia el desarrollo en Android

Los últimos avances en el ámbito del Internet de las Cosas (IoT) nos han permitido controlar diferentes dispositivos utilizando nuestros smartphones. El resultado de esto son hogares inteligentes, edificios inteligentes, transporte inteligente, asistencia sanitaria inteligente y muchas otras innovaciones. Como consecuencia el IoT ofrecerá la posibilidad de desarrollar mas aplicaciones para la gestión de los distintos dispositivos que se verán conectados.

El universo de las aplicaciones móviles es el mayor mercado del sector de la tecnología, con números tan rotundos como los aproximadamente 6,6 millones de aplicaciones móviles presentes en la conocida App Store. La transformación digital es más que una realidad, a finales del año 2017 existían alrededor de 5.000 millones de usuarios de móviles, cada uno de ellos con 15 aplicaciones de media instaladas en sus dispositivos, de las cuales más de la mitad eran de uso diario y continuo.

Con esta realidad social y sus cifras, los desarrolladores de apps se están enfocando en desarrollar aplicaciones según las necesidades del cliente. Una de las grandes razones por las que este nicho de mercado de las aplicaciones móviles está en aumento y se le prevé gran crecimiento en el futuro más cercano.

Entre estas aplicaciones cabe destacar las aplicaciones de bajo demanda, las cuales actúan como un mediador entre los clientes y los proveedores de diferentes servicios. Los usuarios prefieren pagar una pequeña tarifa por un procedimiento más rápido y conveniente que ofrecen este tipo de aplicaciones.

Es muy difícil poner límites a este sector tecnológico que posee uno de los futuros más brillantes de la actual industria tecnológica.

