

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Desarrollo de Aplicación Móvil de Pedidos Proyecto integrador

Lorena Carolina Herrera Cervantes

Ingeniería de Sistemas

Trabajo de titulación presentado como requisito
para la obtención del título de Ingeniera de Sistemas

Quito, 5 de mayo de 2016

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN

Desarrollo de Aplicación Móvil de Pedidos

Lorena Carolina Herrera Cervantes

Calificación:

Nombre del profesor, Título académico

Fausto Pasmay, Ms.

Firma del profesor

Quito, 5 de mayo de 2016

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Lorena Carolina Herrera Cervantes

Código: 00106818

Cédula de Identidad: 1715499826

Lugar y fecha: Quito, 5 de mayo de 2016

AGRADECIMIENTO

Primero, quiero agradecer a mis padres por brindarme su apoyo incondicional y guiarme por el camino correcto. Gracias a ellos no sería la persona que soy hoy en día. También, quisiera agradecer a mi hermana, Estefanía Herrera, por ser una persona especial en mi vida que siempre me ha estado junto a mí en los buenos y malos momentos. Igualmente, quisiera agradecer a Sebastián Sánchez por su apoyo y amor incondicional. Además, quisiera agradecer a mi familia en general por todo el soporte incondicional que siempre me han dado durante todos estos años.

Finalmente, quisiera agradecer a la empresa Herrera Carvajal y Asociados por haber confiado en mi trabajo realizado y brindarme toda la información necesaria para la realización del proyecto.

DEDICATORIA

Este trabajo le dedico de manera especial a mi padre por ser un ejemplo de lucha y perseverancia que siempre ha sido el soporte fundamental de mi vida. Con su apoyo y los recursos brindados pude concluir mis estudios. También se lo dedico a mi madre por todas las enseñanzas que me ha dado a lo largo de mi vida y por darme su amor incondicional.

RESUMEN

El presente trabajo establece el diseño, desarrollo, estructura y funcionamiento de una aplicación móvil de toma de pedidos para el software SAFI ERP. El desarrollo de este trabajo se dio por la necesidad de los clientes de SAFI ERP de tener una aplicación móvil que se encargue de realizar pedidos y cotizaciones instantáneas sin necesidad de recurrir al software de escritorio. Este trabajo fue desarrollado en la plataforma de Android junto con una aplicación de servidor para Java. Esta aplicación de servidor es un intermediario entre la aplicación móvil y el ERP que permite extraer y guardar datos de la base de datos del ERP. Primeramente, se explica la estructura y funcionamiento de la aplicación móvil de toma de pedidos. Dentro de esta sección, se explica cada función de los módulos creados que incluye clientes, productos, pedidos y sincronización. Posteriormente, se explica la estructura y funcionamiento de la aplicación de servidor junto con sus módulos establecidos. Esto incluye el funcionamiento completo de la sincronización con la aplicación móvil, la revisión de cambios y el procesamiento de los pedidos. Finalmente, se explica el funcionamiento completo del proyecto, su sistema de actualizaciones y los resultados obtenidos.

Palabras clave: Android, Aplicación Móvil, Base de Datos, Java, Pedidos, Servidor.

ABSTRACT

This work establishes the development, structure and operation of a mobile application for taking orders integrated with SAFI ERP software. The development of this work was given by the need of SAFI ERP customers to have a mobile application that is responsible for instant ordering without resorting to the desktop software. This work was developed on the Android platform along with an application server for Java. This application server is an intermediary between the mobile application and ERP to extract and store data from the ERP database. First, the structure and operation of the mobile application for taking orders is clarified. Within this section, all the modules created are explained including customers, products, orders and synchronization. Subsequently, it gives details of the structure and operation of the server application together with its established modules. This includes the full operation of synchronization with the mobile application, review of changes and processing of orders. Finally, the full operation of the project is described with the system updates and results.

Key words: Android, Database, Java, Mobile Application, Orders, Server.

TABLA DE CONTENIDO

1. Introducción	11
1.1 Antecedentes	11
1.2. Justificación.....	12
1.3. Objetivos Específicos.....	13
2. Aplicación Movil	17
2.1 Descripción	17
2.1.1 Módulo de Clientes.....	18
2.1.2 Módulo de Productos	18
2.1.3 Módulo de Pedidos.....	18
2.1.4 Módulo de Sincronización.....	19
2.1.5 Módulo de Configuración	19
2.2 Estructura	20
2.2.1 Diseño de Base de Datos.....	21
3. Aplicación de Servidor	23
3.1 Descripción	23
3.1.1 Módulo de Extracción de Datos	23
3.1.2 Módulo de Revisión de Cambios.....	24
3.1.3 Módulo de Envío y Recepción de Datos	24
3.2 Estructura	24
3.2.1 Diseño de Base de Datos.....	28
4. Funcionamiento Aplicación Movil - Servidor.....	30
4.1 Diagrama de Flujo	31
5. Resultados.....	32
6. Conclusiones y Recomendaciones	34
6.1 Conclusiones	34
6. 2 Recomendaciones	35
7. Referencias bibliográficas.....	38
Anexo A: Deficiones.....	39
Anexo B: Encuesta de Safisfacción del Cliente	40
Anexo C: Manual de Usuario	42
Anexo D: Manual de Instalación y Técnico	52

ÍNDICE DE FIGURAS

Figura 1. Funcionalidades de la Aplicación.....	16
Figura 2. Asynctask creadas para ejecución.....	21
Figura 3. Modelo Entidad-Relación de la base de datos del dispositivo móvil.....	22
Figura 4. Modelo de conexión de Sockets Cliente-Servidor.....	26
Figura 5. Diagrama de Sincronización de datos.....	27
Figura 6. Modelo Entidad – Relación base de datos de Servidor.....	29
Figura 7. Funcionamiento completo de Aplicación móvil y Servidor.....	30
Figura 8. Diagrama de Funciones.....	31

ÍNDICE DE TABLAS

Tabla 1. Cumplimiento de lista de tareas de trabajo de titulación.....	32
Tabla 2. Resultado de la Encuesta de Satisfacción al Cliente.....	33

1. INTRODUCCIÓN

Las aplicaciones móviles se han convertido en una herramienta indispensable para los seres humanos, pues han permitido solucionar problemas de la vida diaria, comunicarse e interactuar con el mundo. El tiempo que un usuario pasa en una aplicación móvil al día en Estados Unidos es alrededor de 2,8 horas (Dogtiev, 2015). Existe un sinnúmero de aplicaciones de todo tipo disponibles en el mercado para su uso libre y con licencia que permiten a los usuarios seleccionar la que mejor se adapta a sus necesidades. Por esta razón, las empresas invierten en el desarrollo de aplicaciones móviles. Una aplicación móvil está disponible las 24 horas al día, lo que facilita que esté utilizable siempre, aumente la eficiencia, reduzca costos e incrementen las ventas. El número total de aplicaciones móviles descargadas en el 2015 abarca los 50 billones (Dogtiev, 2015). Por lo tanto, crear aplicaciones móviles transforma el negocio, brinda soluciones a sus usuarios y adquiere nuevos clientes.

Por esta razón, para el trabajo de titulación se desarrolló una aplicación móvil de toma de pedidos para el Software SAFI ERP. Con la aplicación, los clientes pueden realizar una cotización y un pedido y enviar al ERP para ser procesada. Esta funcionalidad adicional permite que los vendedores generen varias cotizaciones al día sin necesidad de recurrir al ERP, se procesen los pedidos, y puedan ser entregados inmediatamente al cliente.

1.1 ANTECEDENTES:

Las aplicaciones móviles dentro de una empresa son muy utilizadas, ya que permiten efectivizar las funciones y mejorar la productividad y rendimiento de los empleados. Por lo tanto, las empresas invierten dinero en aplicaciones móviles que faciliten a los trabajadores a cumplir con su trabajo a tiempo, aumentar su productividad y brindar soluciones para cumplir con las tareas asignadas.

En el mercado ecuatoriano, las empresas se han adaptado a los cambios tecnológicos para innovar su imagen, mejorar su campo productivo, ampliar su audiencia de clientes y competir en el mercado nacional e internacional. En el Ecuador, “se encuentran opciones en negocios como farmacias, ropa, viajes, etc. Ahora las personas quieren hacer todo desde su dispositivo móvil” (Revista Líderes, s.f.). Por lo tanto, las empresas ecuatorianas han crecido y se han adaptado a nuevas tecnologías disponibles hoy en día.

La empresa Herrera Carvajal & Asociados se dedica a la elaboración, programación y venta del Software contable llamado SAFI ERP. Este software es un ERP que brinda soluciones contables relacionadas con finanzas, administración, ventas, inventarios y nómina. Este software está diseñado para ser utilizado en Desktop, por lo que sus clientes han solicitado una aplicación móvil que les permita realizar pedidos de productos y servicios para ser enviados a este programa informático y que puedan ser procesados. Por esta razón, se plantea realizar una aplicación móvil de toma de pedidos y órdenes de compra para los clientes de SAFI ERP. Esto permite a los vendedores realizar cotizaciones y ventas dentro y fuera de su lugar de trabajo y que sean efectivizados en el sistema ERP de manera inmediata.

1.2 JUSTIFICACIÓN:

El trabajo consiste en realizar una aplicación móvil en la plataforma de Android para realizar toma de pedidos de los vendedores hacia los clientes de una empresa. Este tema ha sido seleccionado, ya que existen varios vendedores que realizan sus ventas fuera de las instalaciones de su empresa, por lo que deben llevar sus catálogos para vender a sus clientes. Sin embargo, al momento que el cliente selecciona los productos, el vendedor debe acceder al ERP de su empresa para realizar la cotización de los productos seleccionados. Esto es un problema, ya que no se tiene una cotización inmediata con el precio total de los productos

que el cliente seleccionó. Por esta razón, se planteó el desarrollo una aplicación móvil donde los vendedores de una empresa puedan realizar cotizaciones y pedidos de productos durante una venta a un cliente. Esto es un beneficio para las empresas, ya que los pedidos se efectivizan inmediatamente y las ventas de las empresas crecerían por mayor cantidad de pedidos y ventas oportunas.

Se decidió realizar una aplicación móvil, ya que existen muchos vendedores que no tienen acceso a internet cuando visitan clientes, por lo que la información de los pedidos se guardaría en la base de datos de la aplicación hasta que el usuario tenga acceso a internet y pueda enviar los datos al servidor. Para esto, se realizó un estudio de los vendedores que trabajan con este software y se llegó a la conclusión de que la mayoría de ellos no poseen un acceso a internet durante su jornada de ventas; y, la mejor manera de mantener esos pedidos es que se guarden en la aplicación hasta que se pueda hacer una sincronización. Finalmente, se seleccionó la plataforma de Android, ya que luego de varias investigaciones de la empresa propietaria se determinó que sus clientes con el software SAFI ERP utilizan dispositivos móviles Android. Esto se debe a su bajo costo en comparación con otros dispositivos móviles que utilizan iOS, BlackBerry, Windows Phone etc.

1.3 OBJETIVOS ESPECÍFICOS:

1. Crear una aplicación móvil de toma de pedidos para software SAFI ERP.

Se propone realizar una aplicación móvil de Android para toma de pedidos y una aplicación de servidor en JAVA que se encargue de procesar los datos de los clientes, productos y pedidos y enviar o recibir y guardar en la base de datos de SAFI ERP. La aplicación móvil contiene información de clientes disponibles, productos disponibles y pedidos

realizados por el vendedor en los que puede revisar el estado, es decir, si se encuentran activos o no.

El vendedor tiene la posibilidad de crear clientes para realizar cotizaciones inmediatas sin necesidad de que se creen en el ERP. La aplicación posee un sistema de sincronización de datos para que sean enviados al servidor. Por ejemplo, antes de enviar el pedido, se actualizan los datos con la última versión de la base de datos para verificar los clientes y la disponibilidad de los productos para los pedidos. Con esta sincronización, el dispositivo móvil está siempre actualizado y con los datos existentes reales. Por lo tanto, el proceso de cotización y pedidos se lo procesa de manera rápida y eficaz para permitir a los vendedores efectuar más ventas.

Con esta aplicación, los vendedores de una empresa pueden visitar a sus clientes y realizar una cotización junto con un pedido al momento que el cliente decide los productos que necesita. Con este pedido, los vendedores de la empresa pueden cotizar y generar una factura para que se entregue de manera inmediata el pedido del cliente. Con esta funcionalidad, se estima incrementar los pedidos y ventas realizadas al día y mejorar la operatividad y productividad de la empresa.

2. Crear una aplicación de servidor que se encargue del procesamiento de los datos entre la aplicación móvil y servidor.

Se propone desarrollar una aplicación de servidor que efectúe consultas en la base de datos acerca de la información de los productos, pedidos y clientes. Esta información es procesada y enviada a la aplicación móvil. Asimismo, esta aplicación recibe la nueva información ingresada por la aplicación móvil, procesa, verifica que los productos estén disponibles e ingrese los nuevos pedidos y clientes a la base de datos para que se efectivice en el software SAFI ERP.

Por otra parte, el servidor tiene un proceso de actualización que revisa los nuevos cambios en la base de datos cada 5 minutos. Si el programa detecta una nueva actualización, inmediatamente actualiza la base de datos del servidor y guarda los cambios en un archivo XML para que cuando el dispositivo móvil solicite una sincronización, este pueda ser enviado. Finalmente, maneja versiones de bases de datos que se actualizan cada vez que se realice un cambio en las bases de datos. Por ejemplo, si la base de datos principal del ERP tiene un nuevo cliente ingresado, la aplicación de servidor inmediatamente inserta este cambio en la base de datos, incrementa una versión y guarda los nuevos datos en un archivo XML.

3. Explicar el funcionamiento completo de la aplicación móvil y servidor.

Estas dos aplicaciones, móvil y de servidor, deben estar integradas para que funcione correctamente. El servidor envía la información a la aplicación móvil que está actualizada con la última versión de la base de datos y procesa los pedidos que son guardados en la base de datos del ERP. Este aplicativo es un intermediario entre el ERP y la aplicación móvil para controlar el flujo de datos. Por consiguiente, permite que exista un control y una verificación de los datos que son enviados a la aplicación móvil para evitar datos duplicados o innecesarios que no se requiere para el proceso de pedidos.

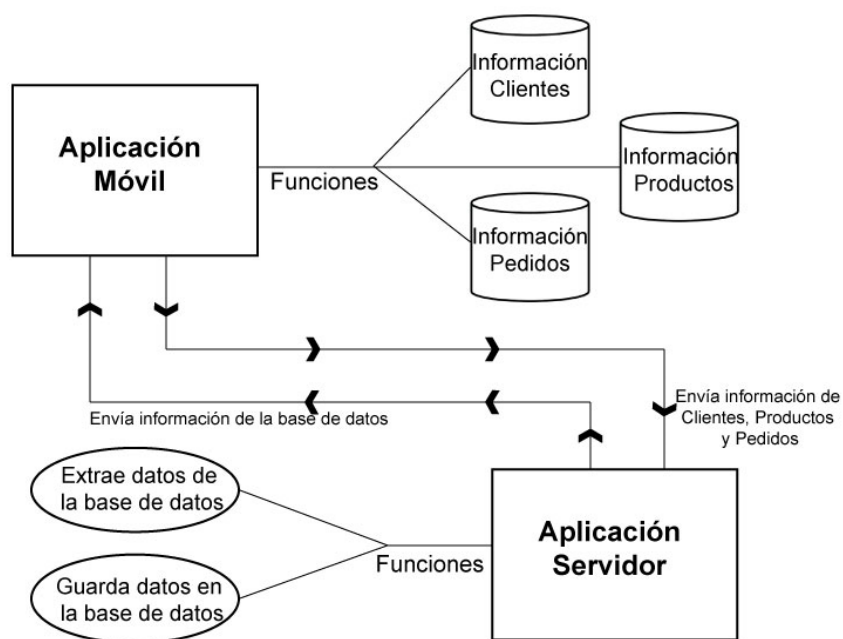


Figura 1. Funcionalidades de la Aplicación

4. Indicar la lista de tareas cumplidas y la evaluación de los clientes.

Para evaluar el funcionamiento de la aplicación, se explica las tareas que se cumplieron durante el desarrollo de la aplicación móvil y de servidor. Con esta información, se puede determinar si se cumplieron con éxito los requisitos del cliente o no. Igualmente, se realiza una evaluación de los clientes que han utilizado la aplicación y su opinión acerca de su funcionamiento.

En la siguiente sección, se describe detalladamente como está estructurada la aplicación móvil, el servidor y su funcionamiento.

2. APLICACIÓN MÓVIL

La aplicación móvil se caracteriza por permitir al usuario realizar pedidos y cotizaciones para sus clientes con la información proporcionada por el ERP. Los datos que se encuentran en la aplicación móvil pueden ser descargados y sincronizados por medio de un sistema de sincronización diseñado para que la aplicación pueda operar de modo offline, en caso que el cliente no posea internet a la hora de utilizarlo. Por lo tanto, para que el aplicativo funcione al 100% debe estar configurado con la dirección IP del servidor. Finalmente, la aplicación móvil está diseñada para vendedores de una empresa. Los vendedores tendrán un usuario y contraseña especial para su uso móvil (punto de emisión móvil).

2.1 DESCRIPCIÓN:

La aplicación móvil está diseñada y desarrollada para funcionar en dispositivos Android. Tales dispositivos pueden ser smartphones, tablets, etc. La versión mínima de Android en la que funciona la aplicación es a partir del API 11 de Android HONEYCOMB. De la misma manera, el dispositivo móvil debe tener una capacidad de almacenamiento de 40MB disponibles para la aplicación. Además, toda esta información se encuentra guardada en una base de datos SQLite diseñada para dispositivos móviles. Finalmente, la aplicación está planteada para funcionar sin conexión a internet. Por lo tanto, el usuario debe descargar los datos con conexión a internet antes de utilizarlos para que sean actualizados y su funcionamiento este garantizado; y, envíe los datos que se guardaron con anterioridad para que sean procesados. La aplicación contiene 4 principales secciones que son clientes, productos, pedidos, sincronización y configuración.

2.1.1 Módulo de Clientes:

En la sección de clientes, se puede observar todos los clientes disponibles que posee la empresa. Es importante recalcar que todos estos datos son extraídos de la base de datos del ERP. Una de las funcionalidades de la aplicación es que se puede observar la información principal de los clientes. Dentro de esta funcionalidad, se puede ver en el mapa la ubicación del cliente para verificación de envío de pedidos. Esta información puede ser editada en caso de que los datos se encuentren incorrectos o se requiera ingresar un nuevo cambio. Finalmente, se pueden crear nuevos clientes en caso de que el cliente no se encuentre registrado en la empresa. Los datos ingresados son validados y guardados para que sean enviados en la siguiente sincronización.

2.1.2 Módulo de Productos:

En la sección de productos, se puede observar los productos que se encuentran disponibles en el ERP junto con la información necesaria. Estos datos pueden ser editados en caso de que se encuentre un dato incorrecto o se requiera algún cambio. Finalmente, se puede crear nuevos productos que luego serán revisados en el ERP.

El objetivo de esta funcionalidad es que el vendedor haga la solicitud de nuevos productos que el cliente solicitó en caso de que no se encuentre dentro del stock. Los productos creados se presentan solo si son aprobados. El usuario de la aplicación (vendedor) puede revisar los productos que creó para tener un control de los productos solicitados.

2.1.3 Módulo de Pedidos

En la sección de pedidos, se puede observar todos los pedidos que ha realizado el usuario junto con el cliente del pedido. En caso de que el pedido este incorrecto, se puede anular el pedido para que no sea procesado en el ERP. Es importante recalcar que el usuario

no puede editar un pedido, solo tiene la posibilidad de crear y anular. Finalmente, el usuario puede crear pedidos para esto debe seleccionar el cliente para que aparezcan los productos con su precio disponible para el cliente.

El usuario puede ingresar y eliminar los productos que desee siempre y cuando cumpla con el stock disponible. También, revisa el pedido y los productos que seleccionó antes de guardar el pedido. Cuando el pedido se guarde, se generará un número de pedido asignado exclusivamente para el usuario.

2.1.4 Módulo de Sincronización

En la sección de sincronización, el usuario recibe y envía datos a la aplicación de servidor para que sean procesados e ingresados en el ERP. Primeramente, envía al servidor la versión de base de datos que posee, este le responde con los datos que han sido actualizados durante su última sincronización. Cuando estos datos han sido recibidos, los nuevos pedidos, modificaciones y creaciones de clientes y productos son enviados al servidor para que este procese.

2.1.5 Módulo de Configuración

Finalmente, en la sección de configuración, el usuario revisa los datos que posee y puede editarlos. Por ejemplo, editar su contraseña, su código de punto de emisión, su descripción y su número de serie. Estos 3 últimos datos son utilizados para generar el número de pedido. Adicionalmente, tiene que ingresar la dirección IP del servidor en la que se realizará la sincronización de datos.

En la siguiente sección, se explica cómo está estructurada de la aplicación.

2.2 ESTRUCTURA:

La aplicación móvil contiene muchas actividades que se crea a medida que el usuario selecciona sus opciones. Por ejemplo, cuando el usuario abre la aplicación, le indica la primera actividad creada que es en la que debe ingresar el usuario y contraseña. Cuando presione el botón de ingresar, se crea la actividad del menú con una lista de opciones para seleccionar clientes, productos y pedidos. Cada una de estas opciones le traslada a una nueva actividad dependiendo de la opción seleccionada.

Cada actividad tiene su propia interfaz diseñada dependiendo de la función que cumpla junto con una barra de menú con sus funciones determinadas. Para la pantalla de clientes, productos y pedidos se utiliza una lista para mostrar los nombres correspondientes. Dentro de esta lista se exponen los nombres de clientes, productos o número de pedido con el cliente dependiendo de la opción seleccionada, así como una barra de menú con la opción de búsqueda y creación de datos.

Para la pantalla de la información, edición y creación de clientes, productos y pedidos se utilizaron TextViews (despliega texto), botones y EditText (permite editar o ingresar texto) para ver los datos, ejecutar una función con un clic en el botón y para ingresar datos respectivamente. Todos estos elementos fueron diseñados en un archivo XML para las vistas y programados en las actividades. Incluso, poseen una barra de menú para editar, aceptar o cancelar la modificación de los datos.

Por otro lado, la aplicación contiene un servicio. Un servicio es un proceso que se ejecuta “en el background de la aplicación” (Gironés, 2011), es decir es un proceso que se realiza sin la necesidad de una interfaz de usuario. Este se utiliza para ejecutar largas operaciones o para brindar cierta funcionalidad mientras se utiliza otra aplicación.

Para la aplicación móvil, se creó un servicio que realiza el proceso de sincronización de los datos. Para esto, se crearon hilos asíncronos independientes que se ejecutan al momento de que el servicio se inicie.

Igualmente, se crea un hilo (AsyncTask) para la descarga de clientes, productos, pedidos, lista de precios y precios disponibles, así como para el envío de datos al servidor (Figura 2). Dentro de este hilo, ejecuta la conexión con el servidor, la recepción de la información y el procesamiento de datos para que guarde en la base de datos; y, el envío de datos en el que se leen los datos de la base de datos, se crea un archivo XML y se envía al servidor. Se informará al usuario cuando se cumplieron las tareas con éxito o si hubo algún error.

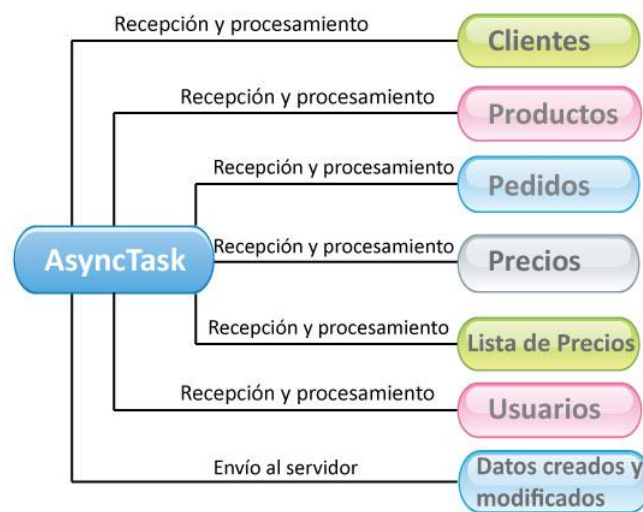


Figura 2. AsyncTask creadas para ejecución.

2.2.1 Diseño de la Base de Datos

Para guardar los datos de los clientes, productos y pedidos se creó una base de datos con SQLite. SQLite es una base de datos relacional que está implementada en la plataforma de Android. Una ventaja de esta base de datos es que no necesita ninguna conexión especial

desde la aplicación. La base de datos contiene 7 tablas distribuidas en clientes, productos, pedidos contiene dos tablas relacionadas que son cabecera y detalle, usuarios, precio y lista de precios (Figura 3). Cada una de ellas contiene las columnas necesarias en las que incluye el identificador. Para el manejo de la base de datos, se creó una clase que maneja la conexión, creación y manipulación de la base de datos con los métodos necesarios para cada una de las funciones de la aplicación.

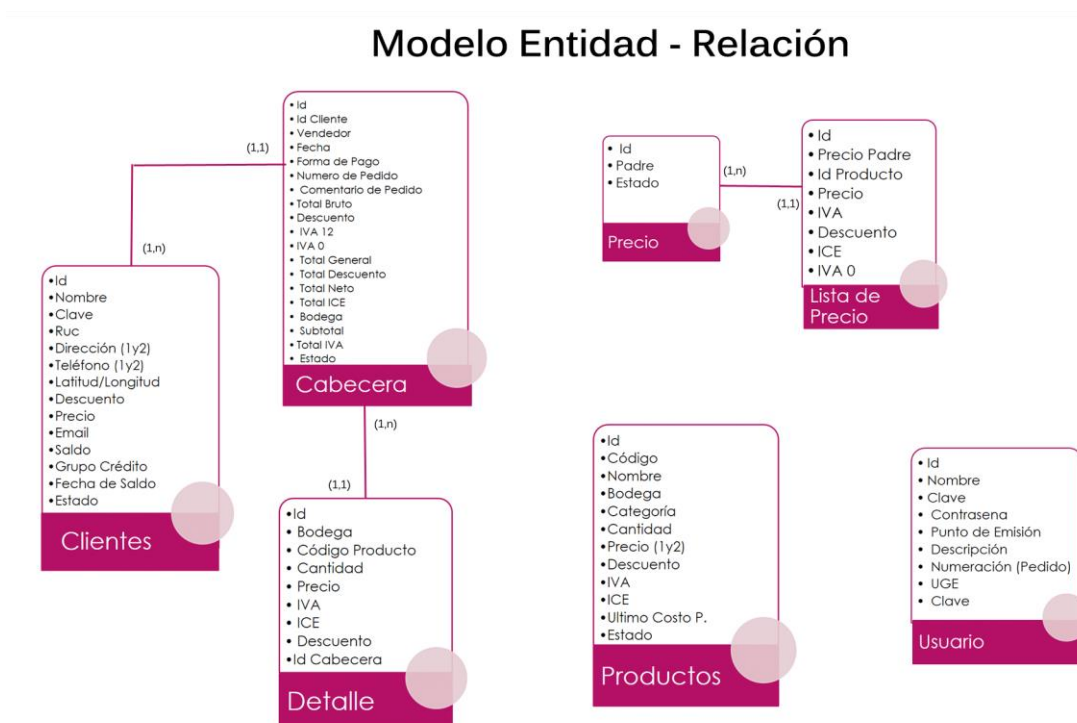


Figura 3. Modelo Entidad-Relación de la base de datos del dispositivo móvil.

3. APLICACIÓN DE SERVIDOR

La aplicación de servidor se caracteriza por procesar los datos y enviar al dispositivo móvil. Primeramente, se encarga de verificar si la base de datos del ERP ha realizado algún cambio en la tabla de clientes, productos, precio, lista de precios y usuarios. Si realizó algún cambio se va a encargar de buscar dicho cambio, insertar en su base de datos y guardar en un archivo XML. Por otro lado, envía los datos al dispositivo móvil, así como procesa los datos recibidos e inserta en la base de datos del ERP para que pueda ser utilizado en dicho software.

3.1 DESCRIPCIÓN:

La aplicación de servidor está programada en la plataforma de JAVA SE. La versión de JAVA utilizada para esta aplicación es JAVA 8. Esta aplicación no contiene una interfaz gráfica implementada, sino que se utiliza en consola. La razón de esta interfaz es que procesa los datos automáticamente sin la necesidad de ingresar ninguna información adicional a parte de nombre, usuario y contraseña de la base de datos. Por lo tanto, la aplicación de servidor consiste en tres principales funciones que son extracción de datos, revisión de cambios y envío y recepción de datos.

3.1.1 Módulo de Extracción de Datos

Su función es extraer los datos necesarios de la base de datos del ERP que incluye información de clientes, productos, pedidos, precio, listas de precio y usuarios e insertar en su base de datos propia. Para esto, se conecta directamente con la base de datos del ERP y realiza consultas con los datos básicos que se necesita para enviar a la aplicación móvil. Una vez extraído estos datos, se guarda en la base de datos del servidor y en un archivo XML. Una

de las razones por las que se guarda en una base de datos propia es para permitir realizar el proceso de revisión de cambios y crear las versiones de la base de datos.

3.1.2 Módulo de Revisión de Cambios

Su función principal es revisar los cambios de la base de datos del ERP junto con la base de datos del Servidor y comparar con los registros ingresados. Si detecta algún cambio realizado, este inserta los datos faltantes en la base de datos carente. Este proceso se realiza cada cierto tiempo y almacena en archivos de versiones. Por ejemplo, si algún cambio fue realizado, este inserta en la base de datos faltante, sube una versión de base de datos y guarda en un archivo XML para ser enviado al dispositivo móvil cuando este lo solicite. Es importante recalcar que estos archivos XML tienen un tamaño máximo establecido que es de 2MB. Esto se debe a que los archivos deben tener un tamaño relativamente pequeño para que los datos se procesen rápidamente en el dispositivo móvil.

3.1.3 Módulo de Envío y Recepción de datos

Su función principal es enviar y recibir los datos de la aplicación móvil dependiendo de las solicitudes recibidas. Es decir que si la aplicación móvil solicita clientes y la versión que posee, el servidor se encarga de enviar los datos de la versión que solicita el dispositivo móvil. Además, si el dispositivo móvil envía una solicitud de envío de datos, el servidor tiene que recibir esos datos, extraer del formato XML e insertar en su base de datos y en la base de datos del ERP.

En la siguiente sección, se explica cómo esta estructura la aplicación del servidor.

3.2 ESTRUCTURA:

La aplicación del servidor se basa en procesos multitarea. Esto se debe a que varios clientes pueden conectarse al mismo tiempo y solicitar alguna información, así como de

revisar cambios en las bases de datos. Para esto, se crearon multihilos que son asignados a cada cliente conectado y al proceso de revisión de cambios.

Para la clase que revisa los cambios en la base de datos, se creó un Timer que ejecuta el código de revisión de cambios de clientes, productos, pedidos, precio, lista de precios, y usuarios cada 5 minutos. Si no se detecta ningún cambio, no se realiza ninguna instrucción. Sin embargo, si se detecta algún cambio, examina si el registro existe en la base de datos del servidor, si no existe entonces crea el dato y si existe lo modifica con los nuevos datos. Cada cambio o creación que se efectúe se almacena en un archivo de XML que será enviado al dispositivo móvil que lo solicite.

Para la clase encargada de conectarse con los dispositivos móviles también se implementó un hilo. Cada vez que se conecte y mande una solicitud un dispositivo móvil, se crea un hilo para que ejecute esa solicitud. Es decir, para recibir o enviar datos. Los hilos se manejan por separado y con cada cliente conectado.

La conexión entre el servidor y la aplicación móvil se realiza mediante sockets TCP. Los sockets son conectores entre computadores que tienen una comunicación continua hasta que uno de los clientes (computadores) cierre la conexión. Para crear un servidor se implementó un socket y un server Socket que debe contener el puerto en el que recibe las peticiones. Con estos dos elementos, se obtiene un sistema de cliente-servidor que recibe y envía datos entre sí (Figura 4).

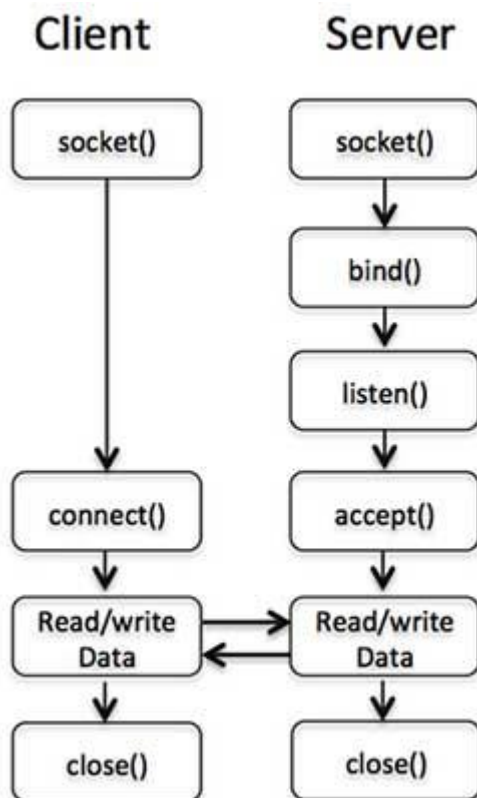


Figura 4. Modelo de conexión de Sockets Cliente-Servidor (Tutorials Point, s.f.).

Para la aplicación de servidor, se implementó un Server Socket que recibe conexión de clientes (aplicación móvil) con el puerto 9001. Cuando recibe una petición, el servidor acepta la conexión y retorna el socket con la información del cliente. Cuando se obtenga la información del cliente mediante el socket, se extraen los datos que envió el cliente. Con los datos recibidos, se procesa la petición (Figura 5).

Por ejemplo, si la aplicación móvil envía una petición de actualización de datos de los clientes, el servidor lee la petición de Clientes con la versión que tiene, con estos datos busca la última versión que posee. En caso de que tengan la misma versión, envía un mensaje informando que las versiones están actualizadas, caso contrario toma los datos de la versión que necesita y envía para que la aplicación móvil procese. En caso de que el pedido tenga

datos que tiene que procesar el servidor, extrae estos datos del XML, los procesa y guarda en la tabla de la base de datos correspondiente.



Figura 5. Diagrama de Sincronización de datos.

Finalmente, en la aplicación de servidor se implementó un sistema de verificación de cambios en la base de datos. Dicha verificación se realiza cada 5 minutos y maneja versiones. Las versiones de tabla de clientes, productos, pedidos, lista de precios, precio y usuarios se guarda en un archivo de texto. Dentro de este archivo se almacena el número de la última versión actualizada de la base de datos y la versión inicial y final del penúltimo archivo creado. Es decir que, si la versión actualizada de la tabla de Clientes es la versión 20, entonces este dato se guarda en el archivo junto con el penúltimo archivo de la versión. Los archivos anteriores a la penúltima versión son borrados automáticamente.

Por ejemplo, si la última versión de la tabla de clientes es 20, el penúltimo archivo puede contener de la versión 15 a la versión 19, por lo que estos datos también se almacenan en

versión inicial y final de clientes. ¿Para qué se utiliza este archivo? Este archivo se utiliza para enviar los datos actualizados a la aplicación móvil. Si la versión móvil posee la versión 14 de clientes, entonces se procede a enviar el penúltimo archivo de versión de clientes junto con el archivo nuevo de actualización. En caso de que el archivo de versión sea menor a las versiones guardadas (nuestro ejemplo menor a 14), entonces se procede a enviar una actualización completa de la tabla de clientes.

3.2.1 Diseño de la Base de Datos

Se creó una base de datos en SQLite que contiene los datos extraídos de la base de datos del ERP. Se utilizó esta base de datos, ya que es una copia exacta a la base de datos que se encuentra en la aplicación móvil. Esta base de datos se creó para verificar los nuevos cambios que se han realizado en la base de datos ERP o en la aplicación móvil para que sean ingresados o modificados según corresponda. La base de datos contiene 7 tablas al igual que la de la aplicación móvil que incluye tabla de clientes, productos, pedidos con su tabla de cabecera y detalle, precio, lista de precios y usuarios (Figura 6).

Modelo Entidad - Relación

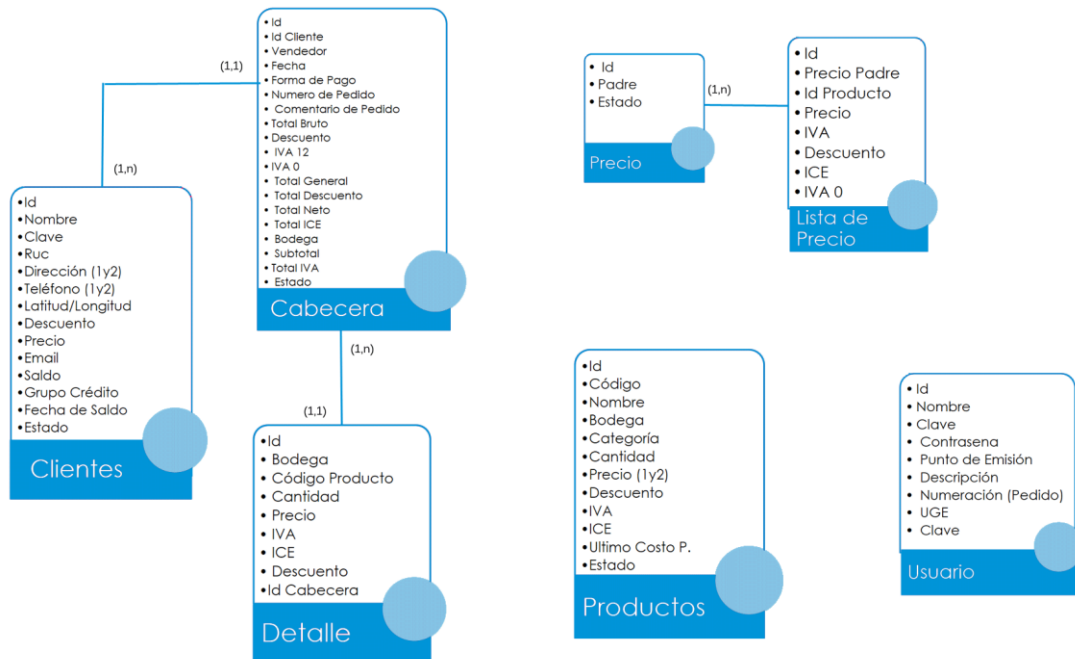


Figura 6. Modelo Entidad – Relación base de datos de Servidor.

A continuación, se presenta una explicación del funcionamiento de la aplicación móvil y el servidor.

4. FUNCIONAMIENTO APLICACIÓN MOVIL - SERVIDOR

Para que la aplicación móvil funcione correctamente, se debe instalar la aplicación de servidor e ingresar el nombre, el usuario y contraseña de la base de datos del ERP. De la misma manera, se debe ingresar la dirección IP del servidor en la aplicación móvil. Estas dos aplicaciones están integradas como se explicó en las secciones anteriores. Su integración se debe al envío y recepción de datos que realiza cada aplicación para su correcto funcionamiento. A continuación, se presenta un gráfico de dicho funcionamiento integrado.

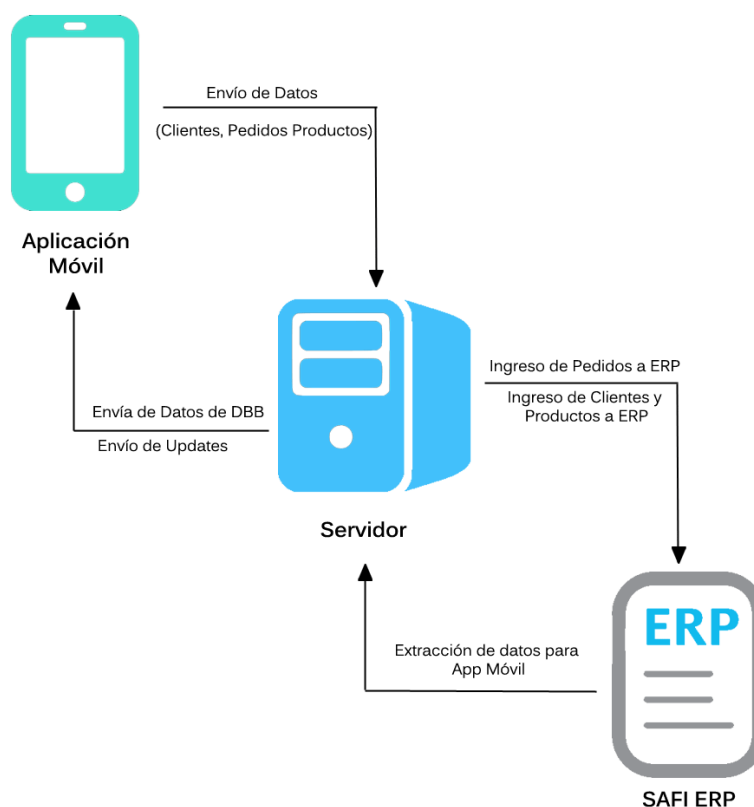


Figura 7. Funcionamiento completo de Aplicación móvil y Servidor.

Como se observa en la figura 7, la aplicación de servidor es el intermediario entre el ERP y la aplicación móvil. Esto se debe a que este aplicativo se encarga de enviar y recibir los datos a la aplicación móvil y de extraer y realizar las respectivas actualizaciones con el ERP. Esta

aplicación siempre debe ejecutarse para garantizar el éxito del funcionamiento de la aplicación en modo offline.

4.1 Diagrama de Flujo

Primeramente, el servidor una vez creado se extrae los datos que necesita la aplicación móvil para su funcionamiento y guarda en su base de datos. Cuando los datos se encuentren ingresados, crea la primera versión de base de datos junto con un archivo XML con los datos que enviará a la aplicación móvil. Posteriormente, verifica modificaciones y almacena en archivos XML y modifica el archivo de versiones. Cuando la aplicación móvil requiera un envío de datos, el servidor envía la base de datos y las actualizaciones (updates).

Cuando la aplicación móvil requiera enviar datos al servidor, esta debe conectarse mediante un socket y enviar una solicitud de envío de datos al servidor. Una vez que el servidor recibe estos datos, el servidor procesa y verifica que los datos se encuentren disponibles y que no se hayan eliminados. Si es que los datos no han sido eliminados, entonces inserta los pedidos y clientes y productos creados o modificados al ERP. Finalmente, el ERP procesa los pedidos y procede a facturar, verifica los productos para que sean aprobados y ver la información de los nuevos clientes.

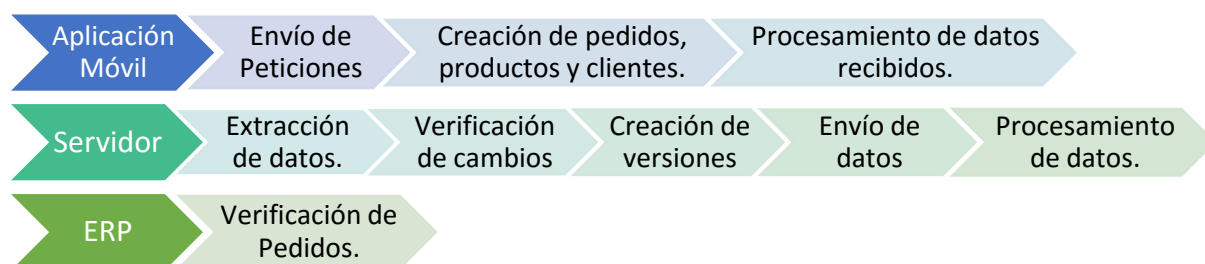


Figura 8. Diagrama de Funciones.

5. RESULTADOS

5.1 Cumplimiento de Lista de Tareas

Antes del desarrollo de la aplicación móvil, se realizó una lista de tareas y requisitos necesarios que la empresa solicitaba para la aplicación. Esta lista contiene las principales tareas que debería realizar la aplicación. A continuación, se muestra una tabla (Tabla 1) con las tareas cumplidas.

Tabla 1. Cumplimiento de lista de tareas de trabajo de titulación.

#	Tareas	Cumplimiento (Si o No)
1	Crear la aplicación móvil de toma de pedidos de Android.	Si
2	Diseñar la interfaz gráfica de la aplicación que incluye los paneles de productos, clientes, pedidos y sincronización de la información.	Si
3	Diseñar la base de datos de la aplicación móvil que va a guardar toda la información de productos, clientes y pedidos.	Si
4	Implementar la interfaz de la sección de productos.	Si
5	Implementar la interfaz de la sección de clientes.	Si
6	Implementar la interfaz de la sección de pedidos y cotizaciones.	Si
7	Implementar la interfaz de la sección de verificación de pedidos.	Si
8	Diseñar la aplicación web en el servidor que se encargue de obtener e ingresar la información a la base de datos del software SAFI y se efectivicen los pedidos.	Si
9	Elaborar las consultas a la base de datos del software SAFI para extraer la información de los productos, clientes y pedidos, y guardar en una base de datos.	Si
10	Elaborar las consultas que van a guardar los datos obtenidos en la aplicación móvil dentro de la base de datos del software SAFI.	Si
11	Diseñar e implementar un sistema de revisión de cambios en la aplicación de servidor.	Si
12	Diseñar e implementar un sistema de actualización de datos.	Si

13	Diseñar e implementar un sistema de sincronización de datos.	Si
14	Realizar pruebas de conexión y funcionamiento entre la aplicación web y el software SAFI.	Si

Como se puede observar en la tabla anterior, todas las tareas establecidas antes de realizar la aplicación fueron cumplidas con éxito. El desarrollo de las dos aplicaciones tomo 3 meses aproximadamente y se realizaron pruebas constantemente para probar su funcionamiento. Por lo tanto, la aplicación se encuentra funcional y ha sido entregada al cliente SAFI ERP para que pruebe su funcionamiento.

5.2 Evaluación del Cliente

La aplicación móvil y servidor fueron entregadas a SAFI ERP para que realice pruebas y pueda utilizar la aplicación y vender a sus potenciales clientes. A partir de su entrega, se ha detectado varios errores que han sido corregidos completamente. La aplicación se encuentra funcional, por lo que SAFI ERP ha comenzado su venta. Se realizó una encuesta a 5 usuarios de la aplicación. A continuación, se muestra una tabla de satisfacción del cliente (Tabla 2).

Tabla 2. Resultados de la Encuesta de Satisfacción del Cliente

Preguntas de Encuesta de Satisfacción del Cliente	Resultado Obtenido (%)
La instalación de la aplicación móvil.	40% Muy Fácil 60% Fácil
La instalación de la aplicación de servidor.	60% Fácil 40% Dificultad Media
La interfaz de la aplicación es fácil de usar	40% Si 60% Probablemente Si
El procesamiento de los datos cuando se sincronización es eficiente	100% Probablemente Si
La creación de clientes, productos y pedidos es rápida	100% Probablemente Si
El envío de datos a SAFI ERP es rápido	100% Probablemente Si
Esta usted satisfecho con el funcionamiento de la aplicación	60% Muy Satisfecho 40% Satisfecho
Cómo calificaría a la aplicación móvil	40% Muy Útil 60% Útil

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- Para desarrollar la aplicación móvil, se analizó las principales funciones que tendría la aplicación. Cada una de ellas fue seleccionada con un objetivo específico. El módulo de clientes se necesita para realizar las cotizaciones y conocer los datos principales. El módulo de productos se utiliza para conocer los productos disponibles y sus características. El módulo de pedidos se utiliza como función principal de la aplicación para realizar cotizaciones. Finalmente, el módulo de sincronización es esencial para el envío y recepción de información.
- Para desarrollar la aplicación de servidor, se realizó un análisis completo de su funcionamiento y conexión con la aplicación móvil. Cada una de sus funciones fue seleccionada para cumplir con el desempeño de la aplicación. La extracción de datos, revisión de cambios y envío y recepción de datos permiten que el sistema cliente-servidor de pedidos trabaje efectivamente.
- El módulo de pedidos de la aplicación móvil permite que se realicen cotizaciones con mayor rapidez y que el cliente pueda conocer el precio de los productos que seleccionó en ese instante. Esto permite que se efectiven los pedidos inmediatamente y se incrementen las ventas.
- La revisión de cambios permite que el sistema aplicación móvil - servidor este actualizado todo el tiempo. Su revisión cada 5 minutos facilita registrar los cambios y modificarlos. Esta funcionalidad proporciona que los datos se encuentren con la última versión de la base de datos y que el usuario pueda utilizarlos sin ningún problema.

- El manejo de versiones provee la actualización y sincronización de datos. Con este archivo, todos los cambios son registrados para luego ser enviados a la aplicación móvil. Este archivo permite que se identifique el número de versión y que el envío de actualizaciones se efectivice de manera rápida. Es importante mencionar que cada vez que exista un cambio en la base de datos, se aumenta el número de versión del archivo.
- La creación de la base de datos en la aplicación de servidor facilita que se puedan identificar y verificar las modificaciones y nuevos ingresos con la base de datos del ERP. Con esto, las bases de datos están actualizadas siempre. Además
- , las bases de datos de la aplicación móvil y servidor son idénticas lo que facilita la sincronización de datos.
- Los problemas que se encontraron durante el desarrollo de la aplicación permitieron que se mejore su diseño y funcionamiento. Cada uno de ellos se analizó y se buscó la mejor solución disponible y efectiva. Con esto, se garantiza que la aplicación móvil es funcional y resuelve un problema para los vendedores.
- La utilización de esta aplicación permitió que los vendedores puedan mejorar su productividad en las ventas, ya que la aplicación permite realizar una cotización rápida de los productos y enviar al ERP para que se procesen. Esto posibilita que los clientes reciban sus productos en el menor tiempo posible y estén satisfechos con el servicio brindado.

6.2 Recomendaciones

- Para el buen funcionamiento de la aplicación, se debe tener una conexión de internet adecuada. Es recomendable sincronizar los datos de la aplicación por medio de Wifi para que los datos se envíen rápidamente y no se consuman muchos recursos.

- El dispositivo móvil que va a contener la aplicación debe tener un espacio de almacenamiento mínimo de 40MB disponibles para que todos los datos sean guardados sin ningún inconveniente y que la aplicación funcione al 100%.
- La aplicación de servidor debe ejecutarse siempre en el servidor desde que se instalada por primera vez. No debe ser cerrada ni eliminada del sistema, ya que la aplicación móvil no podrá funcionar adecuadamente. Si esta deja de funcionar, se debe abrir la aplicación de servidor y ejecutar de nuevo.
- La aplicación de servidor debe ser instala en el servidor donde se encuentre la base de datos del ERP para que se procesen los datos de manera segura. Además, garantiza que la aplicación de servidor trabaje conjuntamente con la base de datos del ERP y pueda ser accedida con la dirección IP externa establecida.
- Se debe realizar una sincronización de datos mínima diaria para que los datos de la aplicación móvil estén actualizados y el stock de los productos sea el adecuado. Esto garantiza que los pedidos sean procesados y aprobados con éxito.
- Se debe realizar una sincronización de datos después de la jornada de trabajo para que los pedidos puedan ser efectivizados ese mismo día. Esto facilita que se entreguen las cotizaciones a tiempo y sean facturadas para proceder al envío de los pedidos.
- Para próximas versiones, la aplicación enviará y recibirá datos cifrados para mejorar la seguridad de la aplicación. Esto evita que los atacantes puedan interceptar los datos y cambiarlos. También los códigos de los productos creados se generarán automáticamente sin necesidad de que el vendedor lo digite manualmente. Finalmente, el procesamiento de los datos durante la sincronización será más rápido y eficiente.

- La aplicación tendrá una sincronización automática cuando detecte una conexión a internet. Con esto, se garantiza que la aplicación pueda actualizarse constantemente.

Esto se realizará en próximas versiones.

7. REFERENCIAS BIBLIOGRÁFICAS

- Brown, C. V., DeHayes, D. W., Hoffer, J. A., Martin, E. W., & Perkins, W. C. (2014). *Managing information technology*. Great Britain: Pearson.
- Christensson, P. (2009, October 27). *Database Definition*. Retrieved 2016, Apr 9, from <http://techterms.com>.
- Christensson, P. (2010, January 4). *Android Definition*. Retrieved 2016, Apr 9, from <http://techterms.com>.
- Christensson, P. (2012, April 19). *Java Definition*. Retrieved 2016, Apr 9, from <http://techterms.com>.
- Dogtiev, A. (2015, December 14). App Usage Statistics: 2015 Roundup - App Industry Insights. Retrieved April 10, 2016, from <http://www.businessofapps.com/app-usage-statistics-2015/>.
- Flores, A. F. (2012). *Programación orientada a objetos usando Java*. Bogotá: Ecoe Ediciones.
- Freier, A. (2015, October 30). WhatsApp revenue and usage statistics - App Industry Insights. Retrieved April 10, 2016, from <http://www.businessofapps.com/whatsapp-revenue-and-usage-statistics/>.
- Gironés, J. T. (2011). *El gran libro de Android*. Barcelona: Marcombo.
- Gupta, S. (2012, December 23). Android Learning Tutorial: Service LifeCycle. Retrieved April 10, 2016, from <http://sunil-android.blogspot.com/2012/12/service-lifecycle.html>.
- Managing the Activity Lifecycle*. (2013, April 5). Retrieved April 10, 2016, from <http://techathon.mytechlabs.com/android-application-architecture-file-structure/> (Originally photographed 2013, April 5).
- Programmer Guru. (2014, January 12). What is AsyncTask in Android? Retrieved April 11, 2016, from <http://programnerguru.com/android-tutorial/what-is-async-task-in-android/>
- Revista Líderes. (s.f.). Los servicios de las aplicaciones se expanden en Ecuador. Retrieved April 10, 2016, from <http://www.revistalideres.ec/lideres/servicios-aplicaciones-expanden-ecuador.html>.
- Tutorials Point. (n.d.). Android SQLite Database Tutorial. Retrieved April 10, 2016, from http://www.tutorialspoint.com/android/android_sqlite_database.htm.
- Tutorials Point. (n.d.). Perl Socket Programming. Retrieved April 10, 2016, from http://www.tutorialspoint.com/perl/perl_socket_programming.htm

ANEXO A: DEFINICIONES

- **Android**: Sistema operativo móvil desarrollado por Google. Se utiliza en varios teléfonos inteligentes. El sistema operativo de Android se basa en el núcleo de Linux (Christensson, 2010).
- **Aplicación Móvil**: Una aplicación de software que se ejecuta en un teléfono inteligente, tableta u otro dispositivo portátil.
- **Base de Datos**: Estructura de datos que almacena información organizada que permite a los usuarios acceder, actualizar y buscar información en base a la relación de los datos almacenados (Christensson, 2009).
- **ERP (Enterprise Resource Planning)**: Conjunto de aplicaciones de negocio integrados, o módulos, que llevan a cabo funciones de negocios comunes (Brown, DeHayes, Hoffer, Martin, & Perkins ,2014).
- **JAVA**: Lenguaje de programación de alto nivel desarrollado por Sun Microsystems. La sintaxis es similar a C++, pero es estrictamente un lenguaje orientado a objetos. Fue diseñado originalmente para el desarrollo de programas para decodificadores y dispositivos de mano, pero más tarde se convirtió en una opción para la creación de aplicación web (Christensson, 2012).
- **SAFI**: software de gestión empresarial corporativo para la planificación de recursos empresariales (SAFI, s.f.).
- **Software**: Conjunto de programas que controlan las operaciones del sistema operativo de ordenador (Brown et al. ,2014).

ANEXO B: ENCUESTA DE SATISFACCIÓN DEL CLIENTE

Evaluación de Satisfacción del Cliente de Aplicación Móvil SAFI Pedidos

1. **La instalación de la aplicación móvil fue:**
 - a. Muy Fácil
 - b. Fácil
 - c. Dificultad Media
 - d. Difícil
 - e. Muy Difícil

2. **La instalación de la aplicación de servidor fue:**
 - a. Muy Fácil
 - b. Fácil
 - c. Dificultad Media
 - d. Difícil
 - e. Muy Difícil

3. **La interfaz de la aplicación es fácil de usar:**
 - a. Absolutamente si
 - b. Si
 - c. A veces
 - d. No
 - e. Absolutamente no

4. **El procesamiento de los datos cuando se sincronización es eficiente:**
 - a. Absolutamente si
 - b. Si
 - c. A veces
 - d. No
 - e. Absolutamente no

5. **La creación de clientes, productos y pedidos es rápida:**
 - a. Absolutamente si
 - b. Si
 - c. A veces
 - d. No
 - e. Absolutamente no

6. **El envío de datos a SAFI ERP es rápido:**
 - a. Absolutamente si
 - b. Si
 - c. A veces
 - d. No

- e. Absolutamente no
- 7. Esta usted satisfecho con el funcionamiento de la aplicación:**
 - a. Muy satisfecho
 - b. Satisfecho
 - c. Normal
 - d. Poco Satisfecho
 - e. Insatisfecho

- 8. Cómo calificaría a la aplicación móvil:**
 - a. Muy útil
 - b. Útil
 - c. Normal
 - d. Inútil
 - e. Totalmente inútil
- 9. Alguna sugerencia para mejorar la aplicación.**

ANEXO C: Manual de Usuario

Descripción de la aplicación:

La aplicación SAFI Pedidos es una aplicación para smartphones Android versión 3.0 o posterior que permite realizar pedidos de productos para que sean procesados en el software SAFI ERP por parte de los vendedores. La aplicación tiene 4 funcionalidades principales que son agregar y editar clientes, agregar y editar productos, realizar pedidos y cotizaciones y sincronizar datos. Para la sincronización de datos, la aplicación debe estar configurada con la dirección IP de la aplicación de servidor, que procesa los datos para el envío o recepción respectivamente.

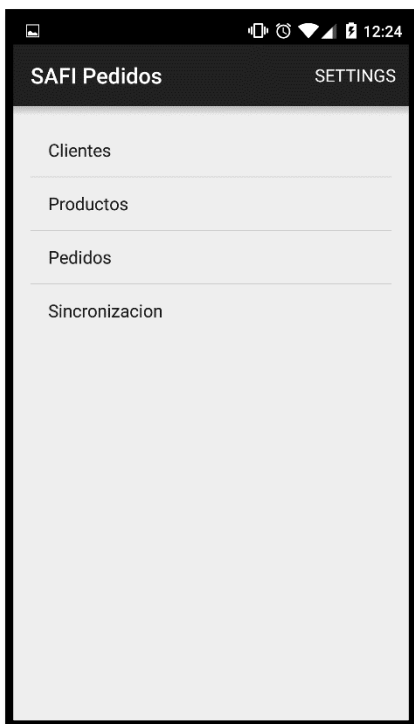
Navegación de la aplicación:

1. Pantalla de Inicio:



Para utilizar la aplicación, debe solicitar un número de usuario y contraseña exclusiva para SAFI Pedidos a SAFI ERP. Con los datos obtenidos, puede ingresar el nombre de usuario y contraseña.

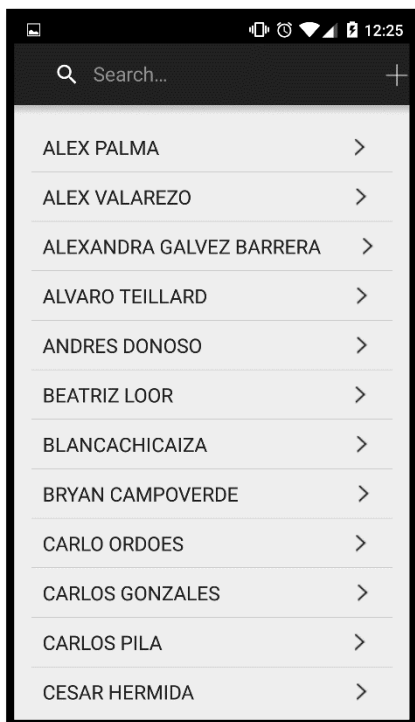
2. Menú Principal



El menú principal tiene 5 opciones principales:

1. Acceso a menú "Clientes" donde se encuentra toda la información de clientes,
2. Acceso a menú "Productos" donde se encuentra toda la información de productos disponibles.
3. Acceso a menú "Pedidos" donde se encuentran los pedidos realizados por el usuario y la creación de pedidos.
4. Acceso a sincronización donde se recibe y envía los datos del servidor.
5. Acceso a menú de configuración (settings) donde se puede modificar

3. Menú Clientes:



El menú principal de Clientes tiene 3 opciones principales:

1. Agregar clientes donde se puede insertar un nuevo cliente.
2. Seleccionar un cliente donde se puede ver la información principal.
3. Buscar donde se busca el nombre del cliente.

a. Agregar Cliente:

Se puede agregar clientes insertando los datos en los campos disponibles con la información del cliente. Para crear el cliente, se debe hacer clic en el botón de visto y para cancelar, en el botón de equis.

Formulario de 'Agregar Cliente' - Datos Generales:

Razón Social:	Nombre del Cliente
Clave:	Clave del Cliente
Ruc:	Ruc del Cliente
Email:	Email del Cliente

Ubicación:

Ciudad:	Ciudad
Código Postal:	Codigo Postal
Latitud:	Latitud
Longitud:	Longitud

Dirección:

Formulario de 'Agregar Cliente' - Dirección, Teléfono, Saldo y Descuento:

Dirección:

Dirección 1:	Direccion del Cliente
Dirección 2:	Direccion 2 del Cliente

Teléfono:

Teléfono 1:	Telefono del Cliente
Teléfono 2:	Telefono del Cliente

Saldo:

Saldo:	Saldo del Cliente
--------	-------------------

Precio:

Precio:	Seleccione Precio...
---------	----------------------

Descuento:

Descuento:	Descuento para Cliente
------------	------------------------

b. Seleccionar y Editar Clientes:

Detalle del Cliente: ALEX PALMA

Clave:	1720077070
Ruc:	1720077070
Email:	alexpalma90@yahoo.com

Ubicación:

Ciudad:	Quito
Código Postal:	
Latitud:	0.0
Longitud:	0.0

Al hacer clic en el nombre del cliente, se ve toda la información principal.

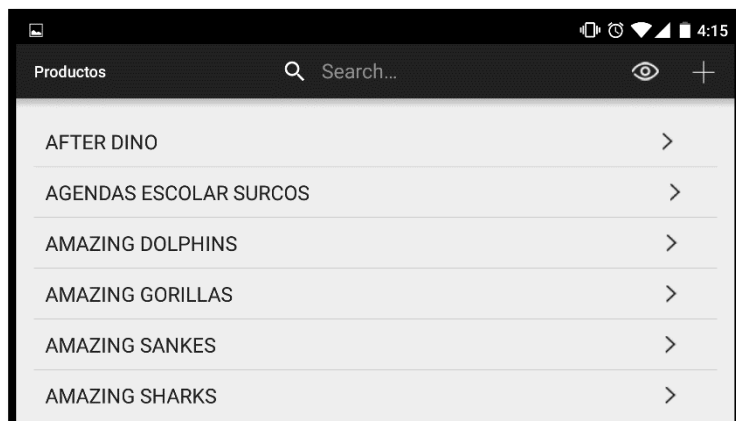
Al hacer clic en el botón de lápiz, se va a poder editar los datos.

En esta pantalla, se puede editar toda la información principal del cliente. Para aceptar los nuevos cambios, se debe hacer clic en el visto y para cancelar los cambios se debe hacer clic en la equis.

4. Menú Productos

En el menú principal de productos, se tiene 4 opciones principales.

1. Agregar Productos donde se puede crear un nuevo producto.
2. Revisar los productos creados por el usuario.
3. Buscar donde se puede buscar un producto con el nombre.
4. Seleccionar un producto donde se puede ver la información principal del producto.



a. Agregar un producto.

Se puede agregar productos insertando los datos en los campos disponibles con su información correspondiente. Para crear el producto, se debe hacer clic en el botón de visto y para cancelar, en el botón de equis.

b. Seleccionar y editar un producto.

Al hacer clic en el nombre del producto, se ve toda la información principal.

Al hacer clic en el botón de lápiz, se va a poder editar los datos.

The screenshot shows a mobile application interface for editing a product. The title bar at the top reads 'Editar Producto' with a checkmark icon on the left and a close icon on the right. Below the title bar is a blue header with the text 'Datos Generales'. The main content area contains several input fields with labels and values:

Nombre de Producto:	AGENDAS ESCOLAR SUR
Código Producto:	0010000000
Código Bodega:	1
Código Alterno:	
Stock Disponible:	20.0

En esta pantalla, se puede editar toda la información principal del producto. Para aceptar los nuevos cambios, se debe hacer clic en el visto y para cancelar los cambios se debe hacer clic en la equis.

5. Menú Pedidos

The screenshot shows a mobile application interface for a menu of orders. At the top, there is a search bar with a magnifying glass icon and the text 'Search...'. Below the search bar is a list of four items, each with a name, a code, and a right-pointing arrow:

ALEX PALMA - 001002000000004	>
CLIENTE DE PRUEBA - 001001000000006	>
MIRIAN CHACHA - 001002000000002	>
SHIRLEY MONCADA - 001002000000003	>

El menú principal tiene 3 opciones principales:

1. Agregar un nuevo pedido.
2. Buscar un pedido por medio del nombre o código del pedido.
3. Seleccionar un pedido y revisar su información.

a. Agregar un Pedido:

Se puede agregar un nuevo pedido insertando la información solicitada. Antes de agregar los productos, se debe seleccionar el cliente haciendo clic en la flecha o en la lupa.

Para agregar los productos, se debe hacer clic en el botón de productos.

i. Agregar productos:

Producto	Cantida d	Preci o	Eliminar
BERENSTAIN PLAY T BALL	2.0	5.06	ELIMINAR

Para agregar un producto, se debe hacer clic en la flecha o en la lupa. Después se debe agregar la cantidad que necesita y hacer clic en el botón de agregar.

Cuando los productos estén seleccionados, se debe hacer clic en el botón de aceptar para regresar al menú anterior, revisar los datos y crear el pedido.

ii. Guardar Pedido:

Cuando ya se agregaron los productos, se puede revisar el total del pedido. Si los datos se encuentran correctos, se puede agregar el producto haciendo clic en el visto o sino puede ser cancelado haciendo clic en la equis.

Agregar Pedido ✓ ✕

Datos del Pedido

Cliente: BLANCACHICAIZA

Clave Cliente: 1708743313

Número Pedido: Numero Pedido

Comentario: _____

Productos

BERNSTAIN PLAY T BALL

AGREGAR PRODUCTO

Costo Total

Total Bruto: 5.06

Agregar Pedido ✓ ✕

BERNSTAIN PLAY T BALL

AGREGAR PRODUCTO

Costo Total

Total Bruto:	5.06
% Dcto. Gral. (Cliente)	0.0
Total Descuento:	0.0
Total Neto:	5.06
Subtotal:	5.06
Tarifa % IVA:	0.0
Tarifa % IVA 0:	5.06
Total IVA:	5.06
Total ICE:	0.0
Total General:	5.06

b. Seleccionar un pedido:

Datos del Pedido

Cliente: ALEX PALMA

Clave Cliente: 1720077070

Número Pedido: 001002000000004

Estado del Pedido: ACT

Comentario: _____

Productos

BERNSTAIN PLAY T BALL

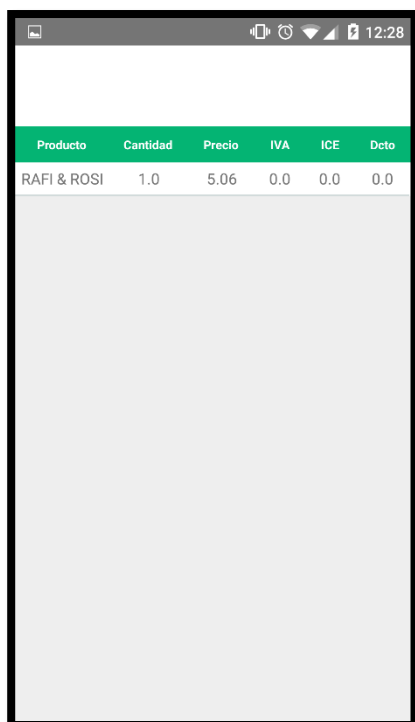
VER PRODUCTOS

Costo Total

Total Bruto:	5.06
% Dcto. General:	8.0
Total Descuentos:	0.0

Al hacer clic en el número de pedido junto con el cliente, se puede ver toda la información del pedido.

Para ver los productos del pedido, se puede hacer clic en ver productos.



Producto	Cantidad	Precio	IVA	ICE	Dcto
RAFI & ROSI	1.0	5.06	0.0	0.0	0.0

Se puede ver los productos del pedido con su precio y cantidad.



BERNSTAIN PLAY T BALL

VER PRODUCTOS

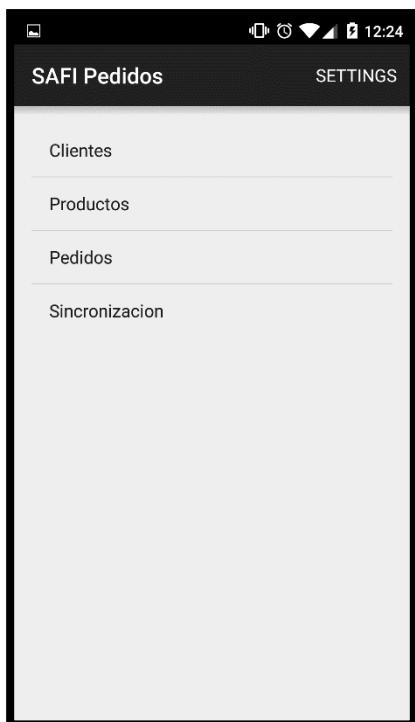
Costo Total	
Total Bruto:	5.06
% Dcto. General:	8.0
Total Descuentos:	0.0
Total Neto:	5.06
Subtotal:	5.06
Tarifa % IVA:	0.0
Tarifa % IVA 0:	5.06
Total IVA:	5.06
Total ICE:	0.0
Total General:	5.06

ANULAR PEDIDO

Se puede ver los datos del precio total del pedido.

Si el pedido se encuentra incorrecto, se puede anular haciendo clic en anular pedido.

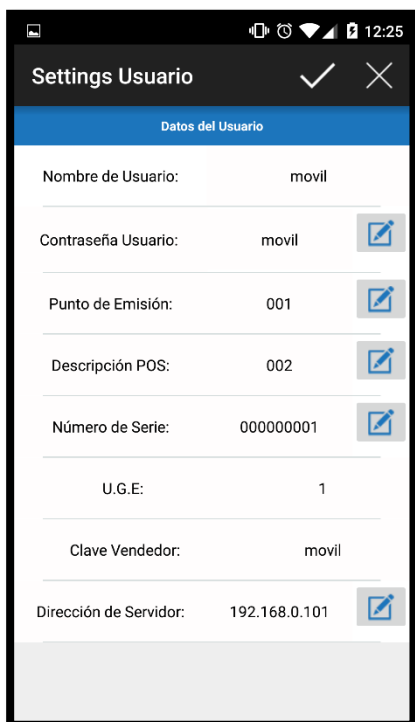
6. Sincronización



Para la sincronización de los datos, se debe hacer clic en sincronización y se procesará automáticamente la petición.

Es importante tomar en cuenta que la dirección del servidor debe estar configurada en configuración (settings) de la aplicación.

7. Menú Configuración:



El menú de configuración permite editar los datos del usuario. Se puede editar la contraseña, punto de Emisión, Descripción, Numero de serie y la dirección del servidor.

Es importante tomar en cuenta que la dirección de Servidor debe estar configurada con la dirección de la aplicación de servidor para que los datos puedan ser sincronizados.

ANEXO D: MANUAL DE INSTALACIÓN Y TÉCNICO

Manual de Instalación

Requisitos para instalar la aplicación móvil (SAFI Pedidos)

A continuación, se presentan los requisitos para instalar la aplicación móvil.

1. La aplicación debe ser instalada en un dispositivo Android con una versión de 3.0 o más.
2. El dispositivo móvil debe tener una capacidad de 40 MB disponible destinadas para el uso de la aplicación y sus datos correspondientes.
3. El dispositivo móvil debe tener conexión a internet para sincronización de datos.
4. El usuario debe solicitar un usuario y contraseña proveniente del ERP para que pueda ingresar.

Requisitos para instalar la aplicación de servidor (SAFI Pedidos)

A continuación, se presentan los requisitos para instalar la aplicación de servidor.

1. La aplicación de servidor debe instalarse en el servidor donde se encuentre instalada la base de datos.
2. Se debe descargar JAVA de <https://www.java.com/es/download/> para que se pueda ejecutar el jar.
3. Debe tener un espacio de almacenamiento para que se guarden los datos del ERP en la base de datos del servidor.

Instalación de Aplicación móvil

1. La aplicación móvil puede ser descargada de <http://safi.com.ec/descargas.php>.
2. Una vez descargada la aplicación, se debe abrir el archivo (apk) y aceptar los permisos.

Instalación y Configuración de Aplicación de Servidor

1. Se debe instalar JAVA.
2. Se debe ejecutar el archivo .jar.
3. Debe ingresar la dirección de la base de datos de SAFI ERP junto con su usuario y contraseña.

Manual Técnico

Aplicación Móvil

La aplicación móvil está programada en Android nativo. Está estructurada con 5 módulos principales que son: Modulo de clientes, productos, pedidos, sincronización y configuración.

Cada módulo posee su clase y sus actividades e interfaces programadas. En este manual se explicada cada una de las clases creadas y su funcionamiento.

#	Nombre de la Clase	Descripción
1	MainActivity.java	<p>Esta clase tiene todos los métodos para la actividad de la aplicación que permite ingresar a la aplicación con el usuario y contraseña. Se encuentra implementado el método onCreate que se ejecuta cuando se llama a esta actividad. En esta clase, se encuentra la programación de toda la interfaz gráfica. La interfaz gráfica de esta actividad se encuentre programada en código XML en el archivo content_main.xml</p> <p>Localización MainActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ProductoActivity.java</p> <p>Localización content_main.xml: PedidosApp\app\src\main\res\layout\content_main.xml</p>
2	ListViewMenuLoader.java	<p>Esta clase tiene todos los métodos para la actividad que indica la lista de opciones que tiene la actividad que incluye clientes, productos, pedidos, sincronización y configuración. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo content_main2.xml.</p> <p>Localización ListViewMenuLoader.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ListViewMenuLoader.java</p> <p>Localización content_main2.xml:</p>

		PedidosApp\app\src\main\res\layout\content_main2.xml
3	<i>PedidosDBHelper.java</i>	<p>Esta clase tiene todos los métodos para la creación y acceso a la base de datos. Esta clase implementa los métodos para insertar y modificar clientes, productos, pedidos, lista de precios y precio.</p> <p>Localización PedidosDBHelper.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\PedidosDBHelper.java</p>

Módulo de Clientes

#	Nombre de la Clase	Descripción
1	<i>Cliente.java</i>	<p>Esta clase contiene toda la información de clientes que incluye variables de nombre, dirección 1 y 2, teléfono 1 y 2, clave, RUC, ciudad, email, longitud, latitud, código postal, saldo del cliente, fecha de creación, grupo de crédito, lista de precio de cliente, descuento de clientes, procesado, y su estado actual. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\Cliente.java</p>
2	<i>Cliente_Activity.java</i>	<p>Esta clase tiene todos los métodos para la actividad de la aplicación que muestra toda la información del cliente y el botón para editar esta información. Se encuentra implementado el método onCreate que se ejecuta cuando se llama a esta actividad. En esta clase, se encuentra la programación de toda la interfaz gráfica. La interfaz gráfica de esta actividad se encuentre programada en código XML en el archivo activity_cliente.xml</p> <p>Localización Cliente_Activity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\Cliente_Activity.java</p> <p>Localización activity_cliente.xml: PedidosApp\app\src\main\res\layout\activity_cliente.xml</p>
3	<i>ClienteAddActivity.j</i>	Esta clase tiene todos los métodos para la actividad que crea un

	<i>ava</i>	<p>nuevo cliente. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_cliente_add.xml</code>.</p> <p>Localización ClienteAddActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ClienteAddActivity.java</p> <p>Localización activity_cliente_add.xml: PedidosApp\app\src\main\res\layout\activity_cliente_add.xml</p>
4	<i>ClienteEditActivity.java</i>	<p>Esta clase tiene todos los métodos para la actividad que edita un nuevo cliente. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_cliente_edit.xml</code>.</p> <p>Localización ClienteEditActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ClienteEditActivity.java</p> <p>Localización activity_cliente_edit.xml: PedidosApp\app\src\main\res\layout\activity_cliente_edit.xml</p>
5	<i>CientesListView.java</i>	<p>Esta clase tiene todos los métodos para la actividad que despliega una lista de todos los clientes. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_clientes_list_view.xml</code>.</p> <p>Localización CientesListView.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\CientesListView.java</p> <p>Localización activity_clientes_list_view.xml: PedidosApp\app\src\main\res\layout\activity_clientes_list_view.xml</p>

6	<i>MapsActivity.java</i>	<p>Esta clase tiene todos los métodos para la actividad que despliega el mapa con las coordenadas de la dirección de un cliente. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad.</p> <p>Localización ClientesListView.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\MapsActivity.java</p>
---	--------------------------	---

Módulo de Productos

#	Nombre de la Clase	Descripción:
1	<i>Producto.java</i>	<p>Esta clase contiene toda la información de productos que incluye variables de nombre, código de bodega, código de producto, código alterno, ultimo costo promedio, precio 1 y 2, descuento, categoría, IVA, iva, procesado y estado. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\Producto.java</p>
2	<i>Producto_Activity.java</i>	<p>Esta clase tiene todos los métodos para la actividad de la aplicación que muestra toda la información del producto y el botón para editar esta información. Se encuentra implementado el método onCreate que se ejecuta cuando se llama a esta actividad. En esta clase, se encuentra la programación de toda la interfaz gráfica. La interfaz gráfica de esta actividad se encuentre programada en código XML en el archivo activity_producto.xml</p> <p>Localización Producto_Activity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ProductoActivity.java</p> <p>Localización activity_producto.xml: PedidosApp\app\src\main\res\layout\activity_producto.xml</p>
3	<i>ProductoAddActivity.java</i>	<p>Esta clase tiene todos los métodos para la actividad que crea un nuevo producto. Esta actividad implementa el método onCreate</p>

	<i>ava</i>	<p>que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_producto_add.xml</code>.</p> <p>Localización ProductoAddActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ProductoAddActivity.java</p> <p>Localización activity_producto_add.xml: PedidosApp\app\src\main\res\layout\activity_producto_add.xml</p>
4	<i>ProductoEditActivity.java</i>	<p>Esta clase tiene todos los métodos para la actividad que edita un producto. Esta actividad implementa el método <code>onCreate</code> que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_producto_edit.xml</code>.</p> <p>Localización ProductoEditActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ProductoEditActivity.java</p> <p>Localización activity_producto_edit.xml: PedidosApp\app\src\main\res\layout\activity_producto_edit.xml</p>
5	<i>ProductoListView.java</i>	<p>Esta clase tiene todos los métodos para la actividad que despliega una lista de todos los productos. Esta actividad implementa el método <code>onCreate</code> que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_productos_list_view.xml</code>.</p> <p>Localización ProductosListView.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ProductosListView.java</p> <p>Localización activity_productos_list_view.xml: PedidosApp\app\src\main\res\layout\activity_productos_list_view.xml</p>

Módulo de Pedidos

#	Nombre de la Clase	Descripción
1	<i>Pedido.java</i>	<p>Esta clase contiene toda la información de productos que incluye variables de identificador del cliente, vendedor, fecha, forma de pago, numero de pedido, comentario del pedido, el total bruto, descuento, IVA 12, IVA 0, total general, total descuento, total neto, total ICE, bodega, productos del pedido, subtotal, total del IVA, estado y procesado. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\Pedido.java</p>
2	<i>INVPrecio.java</i>	<p>Esta clase contiene toda la información de la lista de precios que incluye variables de identificador de precio, identificador de la lista de precio padre, y el estado de la lista de precio. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\INVPrecio.java</p>
3	<i>ListaPrecio.java</i>	<p>Esta clase contiene toda la información de las listas de precio disponibles que incluye variables de identificador de la lista de precio padre, identificador del producto, precio del producto, IVA del producto, IVA 0 del producto, ICE del producto, descuento del producto. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ListaPrecio.java</p>
4	<i>Pedido_Activity.java</i>	<p>Esta clase tiene todos los métodos para la actividad de la aplicación que muestra toda la información del pedido, un botón para ver los productos seleccionados y un botón para anular el pedido. Se encuentra implementado el método onCreate que se ejecuta cuando se llama a esta actividad. En esta clase, se encuentra la programación de toda la interfaz gráfica. La interfaz gráfica de esta actividad se encuentre programada en código XML</p>

		<p>en el archivo <code>activity_pedido.xml</code>.</p> <p>Localización PedidoActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\PedidoActivity.java</p> <p>Localización activity_pedido.xml: PedidosApp\app\src\main\res\layout\ activity_pedido.xml</p>
5	<i>VerProductos.java</i>	<p>Esta clase tiene todos los métodos para la actividad que muestra una tabla con la información de los productos del pedido. Esta actividad implementa el método <code>onCreate</code> que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_ver_productos.xml</code>.</p> <p>Localización VerProductos.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\VerProductos.java</p> <p>Localización activity_ver_productos.xml: PedidosApp\app\src\main\res\layout\ activity_ver_productos.xml</p>
6	<i>AddPedido.java</i>	<p>Esta clase tiene todos los métodos para la actividad que crea un nuevo pedido. Esta actividad implementa el método <code>onCreate</code> que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_add_pedido.xml</code>.</p> <p>Localización AddPedido.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\AddPedido.java</p> <p>Localización activity_add_pedido.xml: PedidosApp\app\src\main\res\layout\ activity_add_pedido.xml</p>
7	<i>AddProductoPedido.java</i>	<p>Esta clase tiene todos los métodos para la actividad que agrega los productos al pedido. Esta actividad implementa el método <code>onCreate</code> que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo</p>

		<p>activity_add_producto_pedido.xml.</p> <p>Localización AddProductoPedido.java: PedidosApp\app\src\main\java\com\example \carolinaherrera93\pedidosapp\AddProductoPedido.java</p> <p>Localización activity_add_producto_pedido.xml: PedidosApp\app\src\main\res\layout\ activity_add_producto_pedido.xml</p>
8	<i>PedidosListView.java</i>	<p>Esta clase tiene todos los métodos para la actividad que despliega una lista de todos los pedidos creados por el usuario. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo activity_pedidos_list_view.xml.</p> <p>Localización PedidosListView.java: PedidosApp\app\src\main\java\com\example \carolinaherrera93\pedidosapp\PedidosListView.java</p> <p>Localización activity_pedidos_list_view.xml: PedidosApp\app\src\main\res\layout\ activity_pedidos_list_view.xml</p>
9	<i>BusquedaClientes.java</i>	<p>Esta clase tiene todos los métodos para la actividad que busca los clientes para insertar en el pedido. Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo activity_busqueda_clientes.xml.</p> <p>Localización BusquedaClientes.java: PedidosApp\app\src\main\java\com\example \carolinaherrera93\pedidosapp\BusquedaClientes.java</p> <p>Localización activity_busqueda_clientes.xml: PedidosApp\app\src\main\res\layout\ activity_busqueda_clientes.xml</p>
10	<i>BusquedaProductos.</i>	<p>Esta clase tiene todos los métodos para la actividad que busca los productos disponibles para el cliente para insertar en el pedido.</p>

	<i>java</i>	<p>Esta actividad implementa el método onCreate que se ejecuta cuando se llama a la actividad. En esta clase, se encuentra toda la programación de la interfaz gráfica de la actividad que se encuentra en código XML en el archivo <code>activity_búsqueda_productos.xml</code>.</p> <p>Localización BusquedaProductos.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\BusquedaProductos.java</p> <p>Localización activity_búsqueda_productos.xml: PedidosApp\app\src\main\res\layout\activity_búsqueda_productos.xml</p>
--	-------------	--

Módulo de Sincronización

#	Nombre de la Clase	Descripción
1	<i>MyService.java</i>	<p>Esta clase tiene todos los métodos para el servicio de la aplicación que se crea el servicio para la sincronización de datos. Se encuentra implementado el método <code>onStartCommand</code> que se ejecuta cuando se crea el servicio. En esta clase, se encuentran clases internas de ejecución de hilos (<code>AsyncTask</code>) para la sincronización de clientes, productos, pedidos, lista de precio, precio y envío de datos.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\MyService.java</p>
2	<i>Servidor.java</i>	<p>Esta clase tiene todos los métodos para la clase que contiene los métodos que se conectan al servidor, envían peticiones, reciben los datos y procesan. Esta clase contiene la conexión con los sockets del servidor. Además, implementa hilos para cada una de las funciones, es decir un hilo para clientes, productos, pedidos, lista de precio y precio respectivamente.</p> <p>Localización VerProductos.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\Servidor.java</p>
3	<i>ServidorSendData.java</i>	<p>Esta clase tiene todos los métodos para la clase que contiene los métodos que se conectan al servidor y envían los datos.</p>

	<p>Esta clase contiene la conexión con los sockets del servidor. Además, implementa hilos para cada una de las funciones, es decir un hilo para clientes, productos y pedidos respectivamente.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ServidorSendData.java</p>
--	---

Módulo de Configuración

#	Nombre de la Clase	Descripción
1	<i>Usuario.java</i>	<p>Esta clase contiene toda la información del usuario que incluye variables de nombre de usuario, contraseña, punto de emisión, descripción, numeración, UGE y clave de vendedor. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\ Usuario.java</p>
2	<i>UsuarioActivity.java</i>	<p>Esta clase tiene todos los métodos para la actividad de la aplicación que muestra toda la información del usuario y las opciones para editar. Se encuentra implementado el método onCreate que se ejecuta cuando se llama a esta actividad. En esta clase, se encuentra la programación de toda la interfaz gráfica. La interfaz gráfica de esta actividad se encuentra programada en código XML en el archivo activity_usuario.xml</p> <p>Localización UsuarioActivity.java: PedidosApp\app\src\main\java\com\example\carolinaherrera93\pedidosapp\UsuarioActivity.java</p> <p>Localización activity_usuario.xml: PedidosApp\app\src\main\res\layout\ activity_usuario.xml</p>

Aplicación Servidor

La aplicación servidor está programada en JAVA. Está estructurada con 5 módulos principales que son: Modulo de clientes, productos, pedidos, sincronización y configuración. Cada

módulo posee su clase con sus métodos programados. En este manual se explicada cada una de las clases creadas y su funcionamiento.

#	Nombre de la Clase	Descripción
1	<i>Cliente.java</i>	<p>Esta clase contiene toda la información de clientes que incluye variables de nombre, dirección 1 y 2, teléfono 1 y 2, clave, RUC, ciudad, email, longitud, latitud, código postal, saldo del cliente, fecha de creación, grupo de crédito, lista de precio de cliente, descuento de clientes, procesado, y su estado actual. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ Cliente.java</p>
2	<i>ConexionBDD.java</i>	<p>Esta clase contiene la conexión a la base de datos del ERP (SQLServer) y las consultas que realiza a la base de datos para extraer la información necesaria. Esta clase utiliza un driver manager para conectarse a la base de datos.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ ConexionBDD.java</p>
3	<i>Pedido.java</i>	<p>Esta clase contiene toda la información de productos que incluye variables de identificador del cliente, vendedor, fecha, forma de pago, numero de pedido, comentario del pedido, el total bruto, descuento, IVA 12, IVA 0, total general, total descuento, total neto, total ICE, bodega, productos del pedido, subtotal, total del IVA, estado y procesado. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ Pedido.java</p>
4	<i>INVPrecio.java</i>	<p>Esta clase contiene toda la información de la lista de precios que incluye variables de identificador de precio, identificador de la lista de precio padre, y el estado de la lista de precio. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\INVPrecio.java</p>
5	<i>ListaPrecio.java</i>	Esta clase contiene toda la información de las listas de

		<p>precio disponibles que incluye variables de identificador de la lista de precio padre, identificador del producto, precio del producto, IVA del producto, IVA 0 del producto, ICE del producto, descuento del producto. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ ListaPrecio.java</p>
6	<i>PedidosDBB.java</i>	<p>Esta clase tiene todos los métodos para la creación y acceso a la base de datos. Esta clase implementa los métodos para insertar y modificar clientes, productos, pedidos, lista de precios y precio.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ PedidosDBB.java</p>
7	<i>Producto.java</i>	<p>Esta clase contiene toda la información de productos que incluye variables de nombre, código de bodega, código de producto, código alterno, ultimo costo promedio, precio 1 y 2, descuento, categoría, IVA, ice, procesado y estado. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ Producto.java</p>
8	<i>Servidor.java</i>	<p>Esta clase se encarga de la sincronización de los datos hacia la aplicación móvil. Contiene una clase interna de ClientesManjeador.java que se encarga de la conexión de cada cliente y el procesamiento de cada petición realizada. Esta es la clase principal de la aplicación de servidor, ya que integra todas las clases creadas.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ Servidor.java</p>
9	<i>ServidorPedidosApp.java</i>	<p>Esta clase se encarga de la extracción de datos de la base de datos del ERP (SQLServer) y de la revisión de cambios en las dos bases de datos. Dentro de esta clase, se implementa el Timer que realiza la revisión cada 5 minutos.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ ServidorPedidoApp.java</p>
10	<i>ServidorThread.java</i>	<p>Esta clase se encarga de crear el socket del servidor para habilitar la conexión con los clientes. Esta clase implementa</p>

		<p>un Thread y el método Run()).</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ ServidorThread.java</p>
11	<i>Usuario.java</i>	<p>Esta clase contiene toda la información del usuario que incluye variables de nombre de usuario, contraseña, punto de emisión, descripción, numeración, UGE y clave de vendedor. También incluye los métodos de acceso para cada una de estas variables.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ Usuario.java</p>
12	<i>XMLClase.java</i>	<p>Esta clase se encarga de leer y crear los archivos XML para envío o recepción de datos. También, se encarga de validar los datos y de guardar en la base de datos del ERP y su propia base de datos. Además, se encarga de crear los archivos con sus respectivas versiones.</p> <p>Localización: ServidorPedidosApp\src\servidorpedidosapp\ XMLClase.java</p>