

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA



“DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE
MASTER EN TECNOLOGÍAS DE LA INFORMACIÓN.
MENCIÓN REDES DE COMUNICACIONES”

TEMA:

“DISEÑO E IMPLEMENTACION DE UNA APLICACIÓN MOVIL A TRAVES DE PEDIDOS ONLINE QUE
INCORPORE TECNOLOGIA RFID PARA GESTION DE INVENTARIO Y PERSONAL EN LOS NEGOCIOS
MINORISTAS DE LA CIUDAD DE QUITO”

AUTOR:

BRYAN ALEXANDER PENAHERRERA CANDO

QUITO 2021/2022

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Peñaherrera Cando Bryan Alexander, portador de la cédula de ciudadanía número 172312209-7, debajo bajo juramento que el presente trabajo de titulación es de mi total autoría, y que se ha respetado las diferentes citas como fuentes bibliográficas que respaldan este trabajo.

AGRADECIMIENTO

A Dios, por ser mi guía en cada paso que doy, por ser mi sustento mi fortaleza y convicción en cada momento de mi vida.

A mi Mama, Abuelita por ser mis pilares en esta vida, por las enseñanzas y momentos que han generado, que yo me encuentre en donde estoy.

A mi tío Wladimir que ha sido como un padre, brindando su apoyo incondicional en el transcurso de mi vida.

A mi esposa Ashly que es mi fortaleza y mi guía en esta vida, ayudándome siempre a sobresalir y brindándome su apoyo incondicional en cada pasito que damos.

A la Pontificia Universidad Católica del Ecuador, por brindarme la oportunidad de continuar con mis estudios en el programa de maestría y compartir sus conocimientos y experiencia.

A mi director Charles Escobar. Ing., Msc. Por su dirección y apoyo en el desarrollo de mi proyecto de titulación.

DEDICATORIA

Dedico esta tesis a las tres mujeres de mi vida mi mamá, mi abuelita y mi esposa, el amor y el cariño que me brindan siempre se ve reflejado en cada paso que doy.

Su apoyo su aliento y las ganas de verme triunfar siempre están ahí. Con mucho agradecimiento y afecto, dedico este esfuerzo para cada una de ellas.

Contents

Contents	IV
1. Introducción.....	1
1.1. JUSTIFICACIÓN	1
1.2. PLANTEAMIENTO DEL PROBLEMA	2
1.3. OBJETIVOS.....	3
1.3.1. OBJETIVO GENERAL.....	3
1.3.2. OBJETIVOS ESPECÍFICOS.....	3
1.4. METODOLOGÍA	4
2. MARCO TEÓRICO Y CONCEPTUAL	5
2.1. ANTECEDENTES.....	5
2.2. MARCO TEÓRICO.....	8
2.3. ALCANCE	13
3. Capítulo 3.....	14
3.1. Llevar el control del stock del inventario de la tienda de abastos.....	14
3.2. Permitir al usuario generar pedidos de manera remota	30
3.3. Diseñar el circuito de lectura de tarjetas con interacción al sistema	42
3.4. Diseño de una app de tipo móvil que integre tecnología RFID para el control de entregas del servicio a domicilio.	46
4. CONCLUSIONES.....	56
5. RECOMENDACIONES.....	56
6. BIBLIOGRAFIA:	57

1. Introducción

“A mediados del siglo XX, las tecnologías de la información y la comunicación (TIC) se integraron diversas áreas de la sociedad para facilitar el desarrollo de tareas complejas. Precisamente, esta es la gran ventaja que aportan las redes informáticas, al permitir la comunicación y el intercambio de archivos a través de aplicaciones que interactúan entre personas.” (Adell,1998)

“La inmediatez, la capacidad para conservar la información, los protocolos de seguridad y el hecho de contar con diferentes opciones para que el usuario pueda conectarse con otras personas son el principal atractivo de este medio, que se ha vuelto tan popular dentro de la sociedad.” (Adell,1998)

Utilizando estas ventajas se pretende utilizar estas herramientas para la creación de un aplicativo móvil que permita llevar el control del inventario de negocios minoristas, además de poder realizar pedidos en línea y llevar el control de las personas que llevan este servicio.

1.1. JUSTIFICACIÓN

A raíz de los acontecimientos que se han venido generando en los últimos años con el avance tecnológico, la interacción de las nuevas plataformas con los usuarios y la alta demanda de un servicio más automatizado e innovador, se han visto en la necesidad de involucrar también con esta tecnología a todo tipo de negocios.

“Los diversos negocios minoristas según la Cámara de Comercio de Quito (CCQ), se han visto con una caída de ventas. Las ventas Privadas, conformadas por el conjunto de facturaciones de Empresas, Personas Naturales y la Economía Popular y Solidaria, al final de 2020 fueron de

USD 134 088 millones, lo que representó un decrecimiento de 16,4%. Esta tendencia continua presente, es así que para el periodo enero – febrero del año 2021, se registra un valor de USD 20 217 millones, lo que representa un decrecimiento de 14,4%.” (Echeveria,2021)

El aplicativo propuesto logrará optimizar y proyectar un aumento y un mayor control del stock que se maneje junto con las entregas que se generen, logrando brindar un servicio de calidad y distinción del negocio.

Tener un mejor acercamiento entre los clientes y el dueño del negocio mediante el aplicativo logrando una relación más personalizada.

Los índices de ventas podrán aumentar en el negocio gracias al control que ejerce el aplicativo, dando un control ordenado de sus productos y balances.

Además, el negocio podrá incrementar el rango de cobertura, gracias a las ventas en línea proyectándose una mayor cantidad de clientes teniendo un costo/ beneficio para la empresa.

Al contar con un aplicativo propio del negocio, el cliente podrá conocer todo lo que ofrece el negocio a tiempo real.

1.2. PLANTEAMIENTO DEL PROBLEMA

En vista de la falta de accesibilidad y disponibilidad de sistemas, hacia los negocios minoristas, estos no cuentan con aplicaciones que les permitan realizar pedidos de sus productos y se encuentren sincronizados con su inventario, además no tienen una forma que también les permita parametrizar el tiempo de envíos de pedidos de manera diaria para que se encargue de gestionar el proceso de entregas.

Además, en vista de la necesidad y la alta demanda de una plataforma propia, los negocios minoristas se enfrentan a otro problema. El no poder tener un control de su stock al generar sus ventas en línea y al no controlar este servicio se generan pérdidas o desfases en sus ventas diarias, limitando incluso el servicio de entregas por la falta de productos en el negocio.

De esto se puede identificar el siguiente problema principal:

- No existe una aplicación móvil a través de pedidos online que incorpore tecnología RFID para gestión de inventario y personal en los negocios minoristas de la ciudad de Quito

Preguntas secundarias:

- ¿Se puede optimizar el control del stock del inventario de la tienda de abastos?
- ¿Es atractivo permitir al usuario generar pedidos de manera remota?
- ¿Es viable diseñar el circuito de lectura de tarjetas con interacción al sistema?
- ¿Es posible Crear una app de tipo móvil que integre tecnología RFID para el control de entregas del servicio a domicilio?

1.3.OBJETIVOS

1.3.1. OBJETIVO GENERAL

- Diseñar e implementar una aplicación móvil a través de pedidos online que incorpore tecnología RFID para gestión de inventario y personal de negocios minoristas de la ciudad de Quito.

1.3.2. OBJETIVOS ESPECÍFICOS

- Llevar el control del stock del inventario de la tienda de abastos.
- Permitir al usuario generar pedidos de manera remota.

- Diseñar el circuito de lectura de tarjetas con interacción al sistema.
- Crear una app de tipo móvil que integre tecnología RFID para el control de entregas del servicio a domicilio.

1.4. METODOLOGÍA

Inicialmente se realizará una investigación bibliográfica para describir las características de la tecnología RFID que permita estar orientado al IoT.

Para la inclusión al IoT se generará un análisis de las características y propiedades del microcontrolador que permita cumplir con los requerimientos y necesidades planteadas.

Se generará una investigación de la productividad que tienen actualmente los micronegocios en el mercado, accediendo a un negocio en particular.

En cuanto a la técnica, se realizará una revisión de textos técnicos, se realizará una revisión de los Datasheet de los diferentes componentes electrónicos, junto con la tecnología RFID. Además, para el presente trabajo de titulación se realizará un diagrama de casos de uso para esquematizar el funcionamiento al generar el sistema.

Sobre las pruebas estandarizadas, son necesarias para observar el funcionamiento del inventario del micronegocio, además, de un modelo Entidad Relación de la base de datos para un análisis del control del inventario e ingreso del producto.

Finalmente, se elaborará un test de rendimiento donde se realizarán pruebas del funcionamiento del sistema en conjunto para verificar la efectividad de la app.

2. MARCO TEÓRICO Y CONCEPTUAL

2.1. ANTECEDENTES

“La tecnología móvil se remonta a los inicios de la Segunda Guerra Mundial, donde ya que era necesaria la comunicación a distancia, es por eso que la compañía que en ese entonces que se conocía como Motorola, creó un equipo llamado Handie Talkie H12-16, este equipo permitía el contacto con sus tropas, pero con la variante que era mediante vía ondas de radio.” (Sietillo99, 2021)

“Dado que la tecnología fue avanzando, se pudo incluir nuevos servicios como EMS y MMS. Logrando con estos un servicio de mensajería, inclusión de música y todo lo relacionado con la multimedia, además de poder enviar todo este tipo de contenido entre dispositivos.” (Sietillo99, 2021)

“El desarrollo de dispositivos trajo consigo un problema, la integración entre plataformas. Por lo tanto, fue necesaria la definición de estándares y protocolos que permitieran una conectividad sin límites, en la que cualquier tipo de software se pudiera ejecutar sobre cualquier plataforma y red. Este fue el punto de partida para el desarrollo de importantes estándares como WAP, UMTS o Bluetooth.” ((Dominguez Mateos, Paredes Velasco, & Santacruz Valencia, 2015))

“Con la aparición de la telefonía móvil digital, fue posible acceder a páginas de Internet especialmente diseñadas para los dispositivos móviles, conocidos como tecnología WAP.” (Sietillo99, 2021)

“Las primeras conexiones se lograban efectuar mediante una llamada telefónica a un número del operador a través del cual se transmitían los datos de manera similar a como lo haría un módem de PC.” (Sietillo99, 2021)

“Posteriormente, nació el GPRS, que permitió acceder a Internet a través del protocolo TCP/IP. Mediante el software adecuado es posible acceder, desde un terminal móvil, a servicios como FTP, mensajería instantánea, correo electrónico, utilizando los mismos protocolos que un ordenador convencional.” (Tecnologiasmovilesenlaeducacionvirtual, 2021)

En nuestro mercado actual se toma en cuenta que el desarrollo de aplicaciones móviles ha cambiado y lo seguirán haciendo.

Las aplicaciones nacen de alguna necesidad y es aquí donde nace la necesidad de solventar estos problemas de forma automatizada. Pero las aplicaciones también pueden responder a necesidades de diversas áreas. “Se suele decir que para cada problema hay una solución, y en informática, para cada problema hay una aplicación.”

“A su vez la tecnología RFID fue evolucionando en conjunto con la tecnología móvil ya que también se dio durante la Segunda Guerra Mundial. Donde se empezó a investigar nuevas técnicas y métodos científicos que puedan dar un giro a favor en la guerra mundial y así tener la ventaja sobre el enemigo, de esta manera vio hacer el mundo los principios básicos de una tecnología inalámbrica; lo que hoy se conoce como RFID.” ((Chang & Lozano, (2013))

“El primer uso de identificación por radiofrecuencia (o RFID) a gran escala se llevó a cabo por británicos, quienes querían ser capaces de distinguir entre sus propios aviones los que regresaban de la costa y los aviones del enemigo. El sistema se desarrolló de la siguiente manera: se colocó un tag en las aeronaves aliadas, al dar la respuesta adecuada a la señal una aeronave propia podía ser distinguida automáticamente de una perteneciente al enemigo.” (Teletronica, 2021)

“Las aplicaciones comerciales de RFID también comenzaron con empresas como Sensormatic y Checkpoint que se fundaron a fines del 1960. Estas empresas, desarrollaron el control electrónico

de artículos (EAS) para contrarrestar el robo. El sistema antirrobo EAS es posiblemente el primer y más extendido uso comercial de la tecnología RFID en el mundo.” (Teletronica, 2021)

Actualmente, la tecnología RFID a crecido y se utiliza para localización, control, almacenaje, inventariado, censo y automatización ya que estas tareas son mucho más eficientes, rápidas y prácticamente libre de errores cuando se sabe utilizar la RFID.

“Una de las principales ventajas del sistema RFID es la de fácil aplicación instalación, montaje y utilización, por esta razón es una de las soluciones más utilizadas actualmente a nivel mundial a prácticamente todo nivel, desde fábricas, librerías, oficinas. Además, es un sistema muy práctico para implementar en casos de que se necesiten sistemas no invasivos o que puedan ser controlados a distancia.” ((Chang & Lozano, (2013))

“Hasta el 2019, las compras a través de Internet movilizan cerca de US\$70.000 millones de dólares anuales, solamente en América Latina y a nivel mundial se prevé que el volumen de ventas será cinco veces mayor en 2020-2021 en los principales mercados.” (Genwords, 2021).

“En Ecuador el 82,3% de las MiPymes utilizan la Internet. La proporción de uso es la siguiente: microempresas 48,6%, medianas 56,9% y pequeñas 52,9%, dando un total general de 52,8%. Sin embargo, a pesar del acceso que tienen a internet se limitan a usarlo para enviar correos y realizar funciones administrativas, solo un 27,4% de las MiPymes tiene presencia en la web.” (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2020).

“Aproximadamente desde hace unos 40 años comenzó el comercio online, pero en la actualidad, sigue creciendo con las nuevas tecnologías, innovaciones y miles de empresas que entran en el mercado en línea cada año. La comodidad, la seguridad y la experiencia del usuario de comercio

online ha mejorado de manera exponencial desde su creación en la década de 1970 hasta hoy.”
(Genwords, 2021)

Queda claro entonces que el comercio online, al estar en constante crecimiento, tiene un futuro prometedor. Ya que todavía tiene un largo camino por recorrer y es una tecnología que falta por entregar todo su potencial.

2.2. MARCO TEÓRICO

El desarrollo de aplicaciones móviles se lo conoce como un conjunto de procesos y procedimientos generados para pequeños dispositivos inalámbricos de cómputo, como teléfonos.

“Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS).” (Oracle, 2021)

“Microsoft Visual Studio es un entorno de desarrollo integrado que permite la generación de aplicaciones móviles con la ventaja de poder crear aplicaciones tanto nativas como híbridas, creado por la compañía Microsoft y está disponible para sistemas operativos Windows, Linux y macOS, y la vez es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP.”(Espacio Honduras, 2021)

“Un Web Service facilita un servicio a través de Internet: generando una interfaz mediante un enlace en el que dos máquinas se pueden comunicar entre sí. Esta tecnología se caracteriza por dos modelos:”

- “Multiplataforma: cliente y servidor no tienen por qué contar con la misma configuración para comunicarse. El servicio web se encarga de hacerlo posible.” (Digital Guide Ionos, 2021)
- “Distribuida: por lo general, un servicio web no está disponible para un único cliente, sino que son diferentes los que acceden a él a través de Internet.” (Digital Guide Ionos, 2021)

“La tecnología RFID es una forma de comunicación inalámbrica entre un lector y un emisor. Semejando su uso con el de un código de barras. De hecho, las etiquetas con esta tecnología son utilizadas tanto en el sector industrial como doméstico. El funcionamiento de esta tecnología es simple. Un receptor envía una señal continua dentro de un radio de alcance concreto. Cuando una etiqueta entra en contacto con esta, lo que hace es enviar información para que el lector interprete según esté programado.” (Universidad Internacional de Valencia, 2017)

Dependiendo de las características de la etiqueta, esta permite grabar la información o editarla, según los requerimientos que se tenga. Teniendo en cuenta esto resulta útil en aplicaciones de control stock o localización de envíos.

“Internet de las cosas (IoT) se centra en la interconexión de la vida cotidiana con la tecnología integrado con sensores, software y otras tecnologías inalámbricas con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet. Los dispositivos que se encuentran van desde objetos domésticos comunes hasta el ámbito industrial.” (Oracle, 2021)

Mediante la informática y tecnologías móviles, las cosas físicas permiten interactuar, con un mínimo de intervención humana. Evitando ciertos errores que se generan por el fallo humano.

“Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos.

Estos procesos o acciones son programados en un lenguaje específico y son introducidos en este a través de un programador. Estos circuitos permiten integrar su tecnología a través de dispositivos móviles teniendo un control sincronizado entre estos componentes.” (Perez, 2004)

Actualmente en el mercado existen varias tecnologías que permite la obtención de la información de un producto hacia un sistema informático. Una de las más usadas es el código de barras ya sea por su costo y su facilidad. No obstante, presenta algunos inconvenientes:

- Son relativamente difíciles de leer: ya que se necesita ubicar correctamente el lector hacia el código para que se pueda obtener una lectura idónea.
- Una vez que son impresos estos no pueden ser modificados.
- Se suelen deteriorar fácilmente por problemas como suciedad. Es aquí cuando, desde hace años se ha a puesto en auge una tecnología por radiofrecuencia (RFID) la cual presenta varias ventajas respecto a los códigos impresos:

Dentro de las cuales se tiene:

- Difíciles de destruir.
- Es fácil de leer. Además, que nos permite leer la información de varios productos al mismo tiempo.
- Permiten almacenar mayor cantidad de información y que nos permite poder modificarla.
- Al momento de generar la lectura por radiofrecuencia, el tag no tiene por qué estar visible.

Principio de Funcionamiento

Para que permita funcionar se necesita realizar siguiente procedimiento: “El lector emite una señal electromagnética la cual al ser recibida por la etiqueta hace que ésta responda mediante otra señal, la cual se envía codificada la información contenida en la etiqueta, logrando realizar el procedimiento de traspaso de información, tal como se ilustra en la Figura 1.” (Perez, 2004)



Campos de Uso

Frecuencias portadoras

“Se debe tomar en cuenta que los sistemas de comunicaciones se encuentran estandarizados por organismos internacionales (UIT, FCC, CCITT, etc.) los cuales se encargan de la regulación, uso y operatividad. Dentro de sus funciones también tienen que repartir el espectro de frecuencia entre todas las aplicaciones que lo necesitan. Con respecto a las aplicaciones de RFID, se han asignado tres bandas principales de frecuencias, las cuales se encuentran en la Tabla 1, la cual contiene las características de los sistemas que usan dicha banda.” (Perez, 2004).

Tabla I. Bandas de frecuencia y usos en etiquetas RFID

Banda de frecuencia	Características	Aplicaciones típicas
Baja 100-500 KHz	Lectura para corta y media distancia Sistemas con tags económicos Velocidad de lectura baja	Control de acceso Identificación de animales Control de existencias Inmovilizadores de automóviles
Intermedia 10-15 MHz	Lectura para corta y media distancia Potencialmente barato Velocidad de lectura media	Control de acceso Tarjetas inteligentes
Alta 850-950 MHz 2.4-5,8 GHz	Lectura para corta y media distancia Velocidad de lectura alta Línea de vista requerida Tecnología cara	Supervisión en sistemas ferroviarios y automotriz. Acceso y control de peaje

Las áreas de aplicación para RFID

Tomar en cuenta que el uso del RFID es muy amplio casi ilimitado en diversos sectores ya que donde existen datos que deben ser leídos o comprobados se puede aplicar este sistema. Las áreas principales de aplicación de RFID son:

- Transporte y logística.
- Domótica.
- Rastreo postal.
- Aeropuertos.
- Pago electrónico.
- Vigilancia electrónica.

“Se basan en un conjunto de herramientas (programas y librerías) que permiten desarrollar que permiten la compilación de programas en lenguaje Java. Existen versiones del JDK para prácticamente todos los Sistemas Operativos y existen también distintos programas comerciales.” (Microsof,2020)

“Los JDK incorporan una herramienta de Debugger (detener la ejecución de un programa en la línea que se desee y poder conocer el valor de las variables en ese momento).”
(Microsof,2020)

2.3. ALCANCE

El presente trabajo de titulación propuesto iniciará con estudios exploratorios debido a que se desarrollara una aplicación móvil a través de pedidos online que incorpore tecnología RFID para gestión de inventario y personal en los negocios minoristas, la programación móvil se encuentra en un auge hace algunos años atrás, la cual se encuentra ocupada por muchas empresas globales pero no se ha logrado introducir de forma específica a micronegocios teniendo detalles específicos que también se puede otorgar de forma accesible y conveniente.

En segundo lugar, se abordará estudios descriptivos puesto que se hará referencia a las características de la tecnología RFID y del IDE de programación de tipo móvil a utilizar, además de los procesos que se deben llevar para sincronizar la base de datos con el sistema; para un funcionamiento conjunto, tomando en cuenta las propiedades que se deben tener, al subir a un sistema de Webservice como lugar de repositorio de la información.

Finalmente, se realizará la entrega de un documento donde conste un diseño de red integrado con un desarrollo para gestionar inventario y RRHH mediante una aplicación que permita generar pedidos de manera online.

3. Capítulo 3

Diseño e implementación una aplicación móvil a través de pedidos online que incorpore tecnología RFID para gestión de inventario y personal de negocios minoristas de la ciudad de Quito.

3.1. Llevar el control del stock del inventario de la tienda de abastos.

Para empezar el proyecto se basa en una serie de requerimientos que se solicitaron los cuales se detallan en los diferentes puntos los cuales permiten cumplir con las necesidades que debe tener la aplicación.

CODIGO DEL REQUERIMIENTO	RF01
NOMBRE	El sistema tiene la capacidad de ingresar y editar el listado de productos que maneje y comercialice el negocio.
DESCRIPCION	El Sistema permite que el usuario ingrese un producto al sistema o lo pueda modificar de ser necesario.
ENTRADA	Categoría del producto (carácter) Nombre del producto. (alfanumérico) Precios/Costos. (numérico) Unidades Disponibles (numérico) Fecha(alfanumérico) Estado(carácter)

SALIDA	Producto ingresado.
PRIORIDAD	Alta

CODIGO DEL REQUERIMIENTO	RF02
NOMBRE	El sistema tiene la capacidad de registrar clientes nuevos.
DESCRIPCION	El Sistema permite que el dueño de la tienda registre a sus clientes para que estos estén atados a la base de datos.
ENTRADA	Nombres y apellidos (carácter) Teléfono (numérico) Numero de teléfono (alfanumérico) Correo electrónico (alfanumérico) Dirección (alfanumérico)
SALIDA	Cliente ingresado.
PRIORIDAD	Alta

No funcionales

Código del Requerimientos	RNF01
Nombre	El sistema debe funcionar hasta un mínimo de 200 usuarios conectados al mismo tiempo.
Descripción	El Sistema permite a 200 o más usuarios hacer uso de forma simultánea.
Entrada	Usuario (alfanumérico) Contraseña (alfanumérico)

Salida	<ul style="list-style-type: none"> • Acceso sin interrupciones a través de la interfaz del sistema • En el caso de que las entradas no sean correctas el sistema mostrara un mensaje que diga “acceso no permitido.
Prioridad	Alta

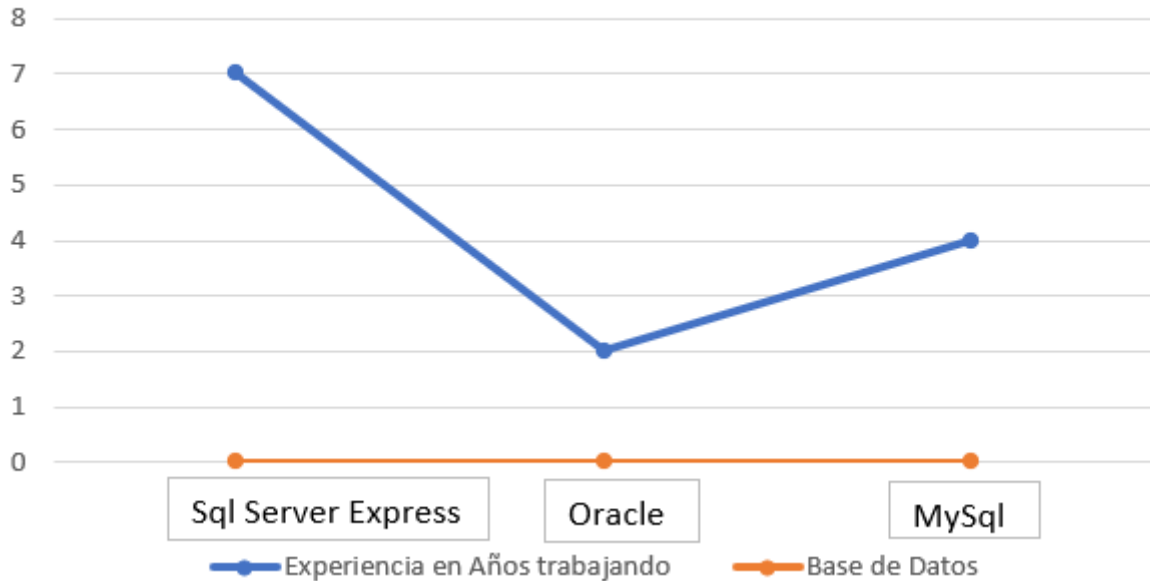
CODIGO DEL REQUERIMIENTO	RNF02
NOMBRE	El Sistema debe permitir que la interfaz responda de forma rápida ante las actualizaciones del Usuario
DESCRIPCION	Los datos ingresados o modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en un tiempo corto no mayor a 5 segundos.
ENTRADA	Ingreso de datos a Interfaz del Usuario. Refrescar la página por parte del Usuario.
SALIDA	Información actualizada en menos de 5 segundos.
PRIORIDAD	Media

CODIGO DEL REQUERIMIENTO	RNF03
NOMBRE	El Sistema debe ser capaz de procesar N transacciones por segundo.
DESCRIPCION	El sistema pueda realizar internamente varias acciones simultaneas al momento de ordenar los productos y encontrar la ubicación de los usuarios.
ENTRADA	Registro de Pedido Registro de Ubicación
SALIDA	Envío y consumo de información
PRIORIDAD	Media

Para el control del inventario se ha basado en una tabla comparativa para la selección de la base de datos donde se va a tener el almacenamiento de esta información.

	Oracle	SQL Express edition	MySQL
Se puede ejecutar en todas las plataformas.	5	5	5
Comunicación con hosting	4	4	4
nos permite el uso de particiones para no perder su integridad en sus datos.	5	4	2
<ul style="list-style-type: none"> No tiene licencias libres. 	0	3	5
Tiene soporte multiplataforma	5	5	5
Puedes crear tantas bases de datos como necesites y también crear varias instancias.	5	5	4
	24	26	25

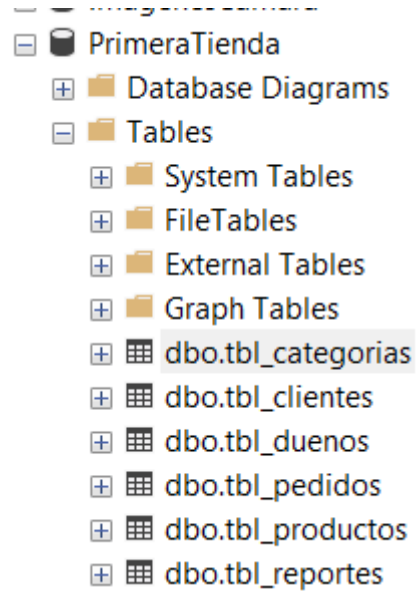
Recta de Aprendizaje



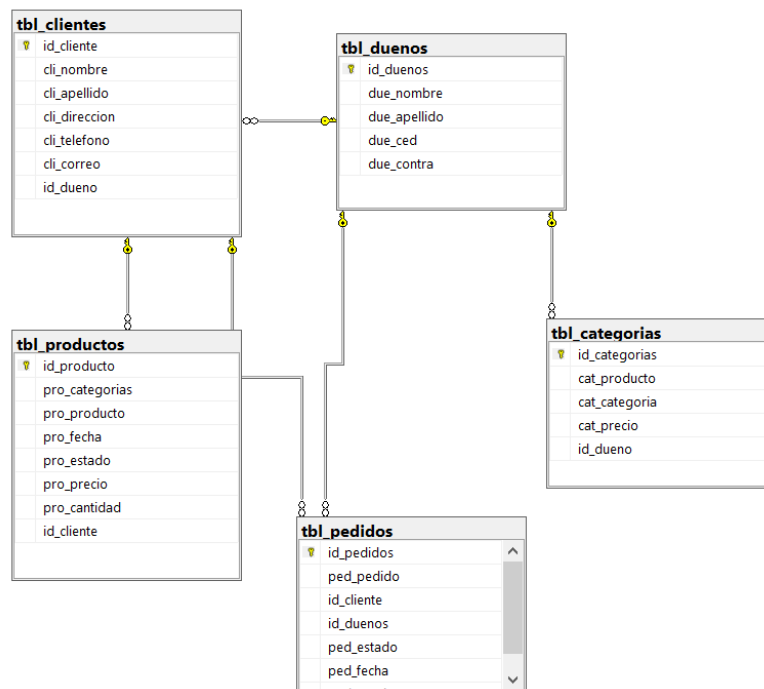
Basándose en la tabla comparativa y la recta de aprendizaje que se desarrollaron se considera que la opción más viable para la elaboración de este proyecto se lo realice mediante SQLSERVER/EXPRESS.

Procedimiento

Para poder almacenar y utilizar toda la información del establecimiento se procedió a crear una base de datos y sus respectivas tablas que tendrán el control y lo necesario para el almacenamiento.



Posteriormente se generó sus respectivas relaciones entre las tablas



Posteriormente se crea las respectivas interfaces que interactúan con la base de datos tanto para almacenar, como para visualizar la información del establecimiento.

•El programa cuenta con un menú para la vista del dueño, en este caso se centra en 2 selecciones que son netamente para la administración y creación de productos que existen en stock dentro del establecimiento el cual se empezara por el añadir producto.



Desarrollo del Código de la vista Indicada

```
pedidos.Click += delegate
{
    Intent intent = new Intent(this, typeof(PedidoBuono));
    // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
    Intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
    StartActivity(intent);
    Toast.MakeText(this, "Seleccione el Pedido", ToastLength.Long).Show();

    //IF (VerificarApp(whatsapp))
    //{
        Intent intent = new Intent();
        intent.SetAction(Intent.ActionSend);
        Intent.PutExtra(intent.ExtraText, "Confirmacion del pedido se lo entregara aproximadamente en min");
        intent.SetType("text/plain");
        Intent.SetPackage(whatsapp);
        StartActivity(intent);
    }
    //else
    //{
        Toast.MakeText(this, "WhatsApp no está instalado. No se pudo enviar el mensaje", ToastLength.Long).Show();
    }
};

productos.Click += delegate
{
    Intent intent = new Intent(this, typeof(AgregarProducto));
    // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
    Intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
    StartActivity(intent);
    Toast.MakeText(this, "Ingrese un Producto", ToastLength.Long).Show();
};

cliente.Click += delegate
{
    Intent intent = new Intent(this, typeof(anadir_personacs));
    // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
    Intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
    StartActivity(intent);
    Toast.MakeText(this, "Ingrese un cliente", ToastLength.Long).Show();
};
```

- Los productos se encuentran clasificados por categoría en el cual se selecciona la categoría deseada y se ingresa el producto.



Vista de la Interfaz

Desarrollo del código de la vista indicada

```
public class AgroagarProducto : Activity
{
    public static readonly string Nombre_usuario;
    @referencias
    protected override void OnCreate(Bundle savedInstanceState)
    {
        WebService1 conex = new WebService1();
        base.OnCreate(savedInstanceState);
        setContentView(Resource.Layout.Productos);
        var nombreusu = Intent.GetStringExtra(Nombre_usuario);
        var bebidas = FindViewById<ImageButton>(Resource.Id.imageButton1);
        var licores = FindViewById<ImageButton>(Resource.Id.imageButton2);
        var confites = FindViewById<ImageButton>(Resource.Id.imageButton3);
        var vegetales = FindViewById<ImageButton>(Resource.Id.imageButton4);
        var panaderia = FindViewById<ImageButton>(Resource.Id.imageButton5);

        licores.Click += delegate
        {
            Intent intent = new Intent(this, typeof(licores));
            // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
            intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
            StartActivity(intent);
            Toast.MakeText(this, "Ingrese un Producto", ToastLength.Long).Show();
        };

        bebidas.Click += delegate
        {
            Intent intent = new Intent(this, typeof(bebidas));
            // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
            intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
            StartActivity(intent);
            Toast.MakeText(this, "Ingrese un Producto", ToastLength.Long).Show();
        };

        confites.Click += delegate
        {
            Intent intent = new Intent(this, typeof(confites));
            // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
            intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
            StartActivity(intent);
            Toast.MakeText(this, "Ingrese un Producto", ToastLength.Long).Show();
        };

        vegetales.Click += delegate
        {
            Intent intent = new Intent(this, typeof(vegetales));
            // intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
            intent.PutExtra(anadir_personacs.Nombre_usuario, nombreusu);
            StartActivity(intent);
            Toast.MakeText(this, "Ingrese un Producto", ToastLength.Long).Show();
        };
    }
}
```

- Luego el usuario llena la información requerida para la tabla del producto y envía la información.



Desarrollo del Código de la vista Indicada

```

1 referencia
public class bebidas : Activity
{
    WebService1 conex = new WebService1();
    public static readonly string Nombre_usuario;
    2 referencia
    protected override void onCreate(Bundle savedInstanceState)
    {
        base.onCreate(savedInstanceState);
        string idduena;
        string categoria = "Bebidas";
        setContentView(Resource.Layout.bebidas);
        var nombreusu = Intent.GetStringExtra(Nombre_usuario);

        var añadir = FindViewById<Button>(Resource.Id.button1);
        var producto = FindViewById<EditText>(Resource.Id.editText1);
        var precio = FindViewById<EditText>(Resource.Id.editText2);
        var cantidad = FindViewById<EditText>(Resource.Id.editText3);

        // Base_datos conex = new Base_datos();
        idduena = conex.ObtenerNombreDuena(nombreusu);

        DateTime fecha = new DateTime();
        añadir.Click += delegate
        {
            AlertDialog.Builder dialog = new AlertDialog.Builder(this);
            AlertDialog alert = dialog.Create();
            alert.SetTitle("Guardando Informacion!");
            alert.SetIcon(Android.Resource.Drawable.AlertlightFrame);
            alert.SetMessage("Esta Seguro de Guardar este Producto");
            alert.SetButton("No", (s, ev) =>
            {
                producto.Text = "";
                precio.Text = "";
                cantidad.Text = "";
            });
            alert.SetButton2("Si", (s, ev) =>
            {
                if (producto.Text=="||| precio.Text=="")
                {
                    Toast.MakeText(this, "Falta llenar los campos", ToastLength.Long).Show();
                }
                else
                {
                    conex.IngresarBebidas(categoria, producto.Text, fecha.Date, Convert.ToDouble(precio.Text), Convert.ToInt32(idduena), Convert.ToInt32(cantidad.Text));
                    producto.Text = "";
                    precio.Text = "";
                    cantidad.Text = "";
                }
            });
        });
        alert.Show();
    }
}

```

* Se selecciona en el menú principal y en el modificar el producto nos desplazara a una vista donde se encuentra clasificada la información y nos visualiza la cantidad que se tiene en stock, además al modificar las unidades o dar de baja algún producto que ya no conste en el establecimiento.



Desarrollo del Código de la vista Indicada

```
using Android.Content;
using Android.Content.PM;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Fast_Market.com.somee.tiendaFinal.www;

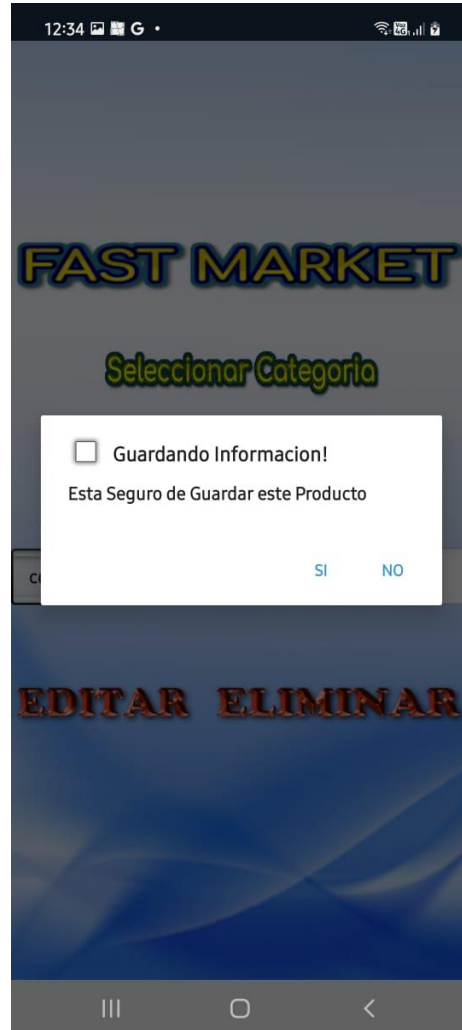
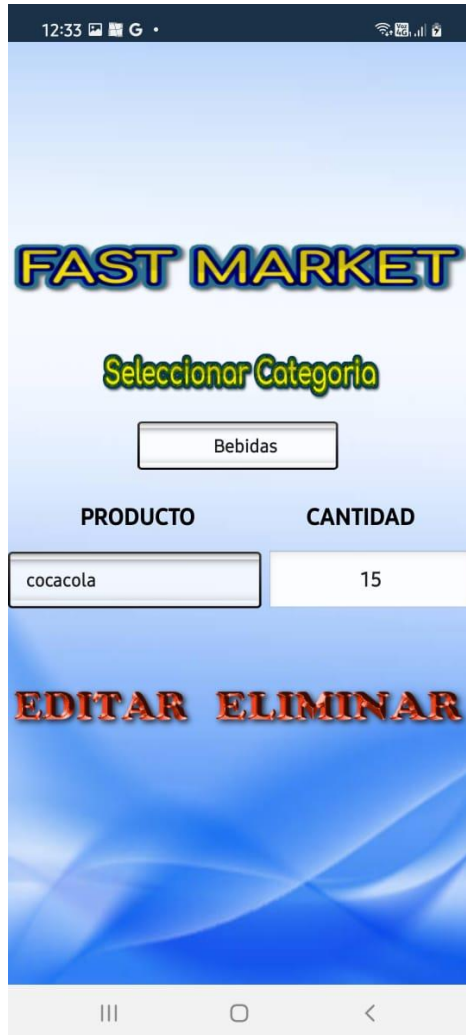
namespace Fast_Market
{
    [Activity(Label = "SeleccionDuena")]
    public class SeleccionDuena : Activity
    {
        WebService1 conex = new WebService1();
        string whatsapp = "com.whatsapp";
        public static readonly string Nombre_usuario;
        bool VerificarApp(String ur1)
        {
            try
            {
                Application.Context.PackageManager.GetPackageInfo(ur1, PackageInfoFlags.Activities);
                return true;
            }
            catch (PackageManager.NameNotFoundException e)
            {
                return false;
            }
        }
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.VistaDuena);
            var nombrousu = Intent.GetStringExtra(Nombre_usuario);
            var productos = FindViewById<Button>(Resource.Id.textView4);
            var cliente = FindViewById<Button>(Resource.Id.textView2);
            var reportes = FindViewById<Button>(Resource.Id.textView1);
            var pedidos = FindViewById<Button>(Resource.Id.textView3);

            var modificar = FindViewById<Button>(Resource.Id.button2);
            var modificar2 = FindViewById<Button>(Resource.Id.button3);

            modificar.Click += delegate
            {
                Intent intent = new Intent(this, typeof(Productos_Muchos));
                // Intent.PutExtra(Seleccion.Nombre_usuario, usu.Text);
                Intent.PutExtra(anadir_personas.Nombre_usuario, nombrousu);
                StartActivity(intent);
                Toast.MakeText(this, "Selecione el Pedido", ToastLength.Long).Show();
            };
        }
    }
}
```

* Al generar la información y los eventos que se desee generar se realizar la consulta o modificación pertinente de los productos.



Desarrollo del Código de la vista Indicada

```

public class Productos_Hechos : Activity
{
    public static readonly string Nombre_usuario;
    List<string> cosas = new List<string>();
    List<string> lista = new List<string>();
    List<string> lista2 = new List<string>();
    string cantidadesf;
    string whatsapp = "com.whatsapp";
    List<string> lista3 = new List<string>();
    List<double> precio = new List<double>();

    List<string> productos = new List<string>();
    List<string> cantidades = new List<string>();

    WebService1 conex = new WebService1();
    0 referencias
    protected override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);

        SetContentView(Resource.Layout.ProductosHechos);
        var nombreusu = Intent.GetStringExtra(Nombre_usuario);

        cosas.Add("Bebidas");
        cosas.Add("Licores");
        cosas.Add("Confites");
        cosas.Add("Panaderia");
        cosas.Add("Vegetales");
        var categoria = FindViewById<Spinner>(Resource.Id.spinner1);
        var listas = FindViewById<Spinner>(Resource.Id.spinner2);
        var editar = FindViewById<Button>(Resource.Id.button1);

        var eliminar = FindViewById<Button>(Resource.Id.button2);
        var cantidadesfinales = FindViewById<EditText>(Resource.Id.editText1);

        var adapterS1 = new ArrayAdapter<this, Android.Resource.Layout.SimpleSpinnerItem, cosas>;
        categoria.Adapter = adapterS1;

        var adapterS1 = new ArrayAdapter<this, Android.Resource.Layout.SimpleSpinnerItem, cosas>;
        categoria.Adapter = adapterS1;
        conex.ObtenerNombreBuenaCliente(Convert.ToInt32(nombreusu));
        categoria.ItemSelected += delegate
        {
            lista.Clear();

            DataSet ds = conex.TraerLista(categoria.SelectedItem.ToString());

            int numerop = ds.Tables[0].Rows.Count;
            string[] vector = new string[numerop];

            for (int i = 0; i < numerop; i++)
            {
                lista.Add(ds.Tables[0].Rows[i]["pro_producto"].ToString());
                precio.Add(Convert.ToDouble(ds.Tables[0].Rows[i]["pro_precio"].ToString()));
            }

            var adapterS11 = new ArrayAdapter<this, Android.Resource.Layout.SimpleSpinnerItem, lista>;
            listas.Adapter = adapterS11;
        };

        listas.ItemSelected += delegate
        {
            lista3.Clear();

            DataSet ds5 = conex.TraerCantidad(listas.SelectedItem.ToString(), Convert.ToInt32(conex.IdBuenaF(Convert.ToInt32(nombreusu))));

            int numerop2 = ds5.Tables[0].Rows.Count;
            string[] vector = new string[numerop2];

```


CODIGO DEL REQUERIMIENTO	RF03
NOMBRE	El Sistema tiene la capacidad al momento de realizar pedidos de restringir productos que ya no se tenga en stock.
DESCRIPCION	El sistema permite llevar un control del stock de productos, lo cual permite visualizar si se cuenta con un producto o ya no al momento de generar el pedido.
ENTRADA	Pedido (alfanumérico) Cantidad (carácter)
SALIDA	Visualización de productos.
PRIORIDAD	Alta

CODIGO DEL REQUERIMIENTO	RF04
NOMBRE	El Sistema tiene la capacidad de almacenar pedidos generados por el usuario.
DESCRIPCION	El sistema permite llevar un control de los pedidos que se generen y visualizar que pedidos ya fueron hechos y cuáles no.
ENTRADA	Pedido (alfanumérico) Estado (carácter) Fecha (carácter) Precio (numérico)
SALIDA	Almacenamiento y visualización de pedidos.
PRIORIDAD	Alta

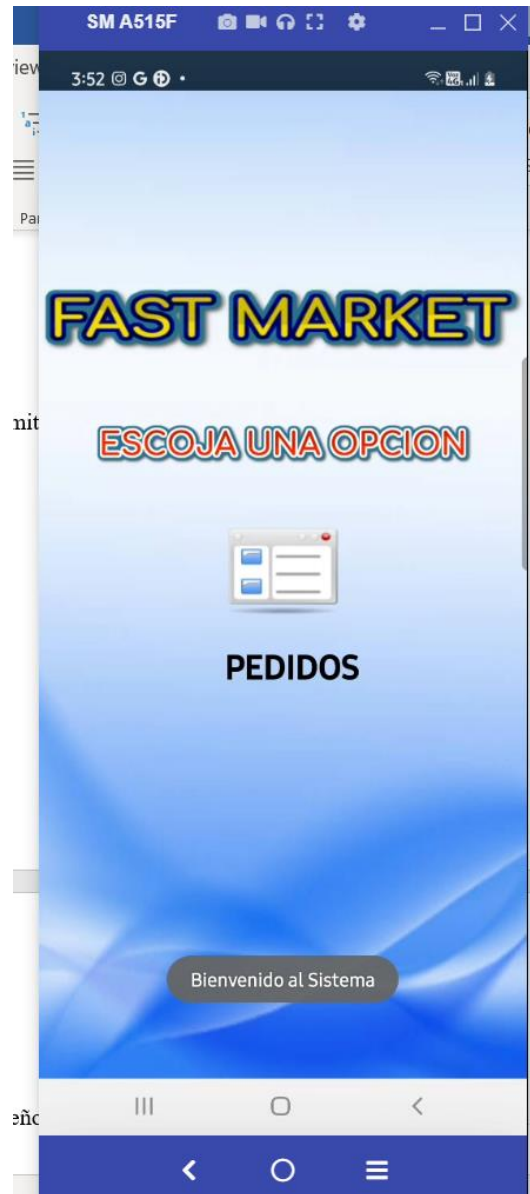
Código del Requerimientos	RF05
Nombre	El sistema tiene la capacidad de notificar los pedidos realizados.

Descripción	El sistema notifica vía correo electrónico y WhatsApp al momento que un cliente genera un pedido.
Entrada	Numero Dueño(alfanumérico) Correo(alfanumérico) Mensaje(alfanumérico)
Salida	Recibir Notificaciones.
Prioridad	Media Alta

Código del Requerimientos	RF06
Nombre	El sistema tiene la capacidad de mostrar las coordenadas de ubicación para el envió.
Descripción	El sistema obtiene las coordenadas de donde se está generando el pedido para poder enviar dicha información.
Entrada	Latitud (numérico). Longitud (numérico).
Salida	Recibir dirección del usuario.
Prioridad	Alta

Para poder realizar este tipo de procedimientos señalados se debe seguir una serie de pasos los cuales se encuentran en las diferentes vistas de la app, las cuales cumplen una función en específico.

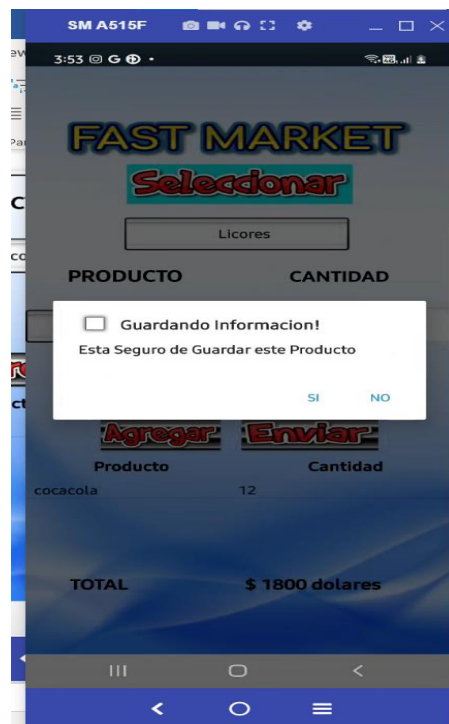
Interfaz principal que permite el acceso a los pedidos.



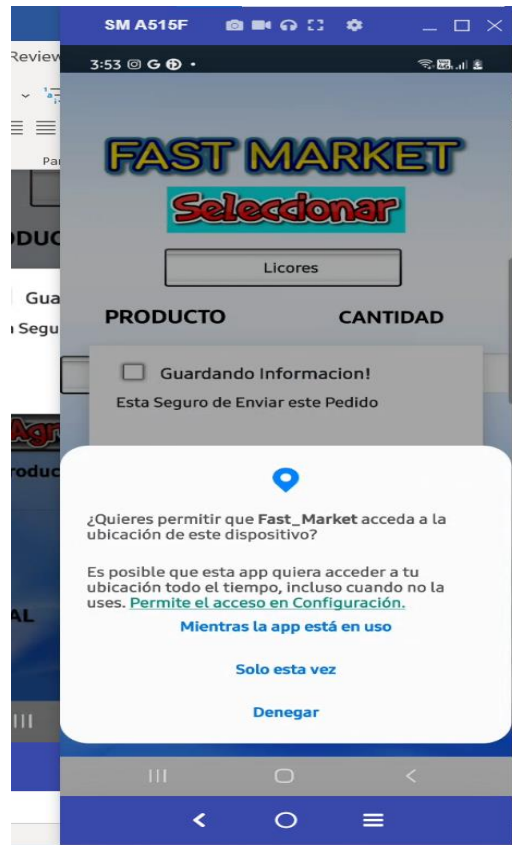
Interfaz para generar pedido clasificado por categorías



El programa solicita cada vez que ingrese un nuevo producto al carrito de comprar se acepte.



Al seleccionar enviar todo el carrito, se activa la ubicación en donde se encuentre.



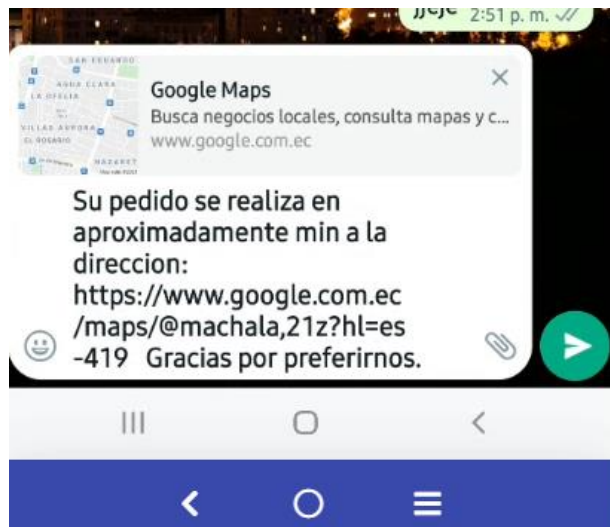
Una vez se envíe el pedido, se habilita el correo y se manda una notificación.



La vista del dueño, en la sección de pedidos estará cargada la información donde tendrá los productos y la dirección en donde debe entregar.



Al dar en confirmar se abre el WhatsApp con el numero registrado y se envía un mensaje



Código:

```

Fast Market | Fast_Market.VistaCliente | OnCreate(Bundle savedInstanceState)
31 {
32     [Activity(Label = "VistaCliente")]
33     public class VistaCliente : Activity
34     {
35         public static readonly string Nombre_usuario;
36         List<string> cosas = new List<string>();
37         List<string> lista = new List<string>();
38         string whatsapp = "com.whatsapp";
39         List<string> lista2 = new List<string>();
40         List<double> precio = new List<double>();
41
42         List<string> productos = new List<string>();
43         List<string> productosactu = new List<string>();
44         List<string> productosactu2 = new List<string>();
45         List<string> cantidades = new List<string>();
46
47         Webservice1 conex = new Webservice1();
48         // Base_datos conex = new Base_datos();
49         double precios;
50         // string format = @"M/d/yyyy H:m:s zzz";
51
52         DateTime fechah = DateTime.Now;
53
54
55         // DateTime fechah2;
56
57
58
59

```

```

References | VistaCliente.xml | AgregarProductos | Reportes | Vegetales | VistaCliente.cs | ProductosRecursos
Fast_Market.VistaCliente | OnCreate(Bundle savedInstanceState)
0 referencias
bool VerificarApp(String uri)
{
    try
    {
        Application.Context.PackageManager.GetPackageInfo(uri, PackageManager.PackageInfoFlags.of(0));
        return true;
    }
    catch (PackageManager.NameNotFoundException e)
    {
        return false;
    }
}
0 referencias
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);

    var nombreusu = Intent.GetStringExtra(Nombre_usuario);
    cosas.Add("Bebidas");
    cosas.Add("Licores");
    cosas.Add("Confiteria");
    cosas.Add("Panaderia");
    cosas.Add("Vegetales");

    int contadorf=0;
    SetContentView(Resource.Layout.VistaCliente);
    double contar;
    string pedi;
    var categoria = FindViewById<Spinner>(Resource.Id.spinner1);

```

```

ence.cs VistaDuena.xml AgregarProducto.cs Reportes.cs vegetales.cs VistaCliente.cs VistaCliente.cs Productos_Hechos.cs
Fast_Market.VistaCliente
OnCreate(Bundle savedInstanceState)

cosas.Add("Panaderia");
cosas.Add("Vegetales");

int contadorf=0;
SetContentView(Resource.Layout.VistaCliente);
double contar;
string pedi;
var categoria = FindViewById<Spinner>(Resource.Id.spinner1);
var listas = FindViewById<Spinner>(Resource.Id.spinner2);
var agregar = FindViewById<Button>(Resource.Id.button1);
var enviar = FindViewById<Button>(Resource.Id.button2);

var listado = FindViewById<ListView>(Resource.Id.listView1);
var listado2 = FindViewById<ListView>(Resource.Id.listView2);

var vprecios = FindViewById<TextView>(Resource.Id.textView5);
var totalp = FindViewById<TextView>(Resource.Id.textView4);
var cantidad = FindViewById<EditText>(Resource.Id.editText1);
contar = 0;

var adapter51 = new ArrayAdapter<this, Android.Resource.Layout.SimpleSpinnerItem, cosas>;
categoria.Adapter = adapter51;

//listado.ItemSelected += delegate
// {
//     string datos;
//     //datos = listado.SelectedItem.ToString();

//     Toast.MakeText(this, datos, ToastLength.Long).Show();

```

```

//});

conex.ObtenerNombreDuenaCliente(Convert.ToInt32(nombreusu));
categoria.ItemSelected += delegate
{
    lista.Clear();

    DataSet ds= conex.TraerLista(categoria.SelectedItem.ToString());

    int numerop = ds.Tables[0].Rows.Count;
    string[] vector = new string[numerop];

    for (int i = 0; i < numerop; i++)
    {
        lista.Add( ds.Tables[0].Rows[i]["pro_producto"].ToString() );
        precio.Add( Convert.ToDouble(ds.Tables[0].Rows[i]["pro_precio"].ToString() ));
    }

    var adapter511 = new ArrayAdapter<this, Android.Resource.Layout.SimpleSpinnerItem, lista>;
    listas.Adapter = adapter511;

};
listas.ItemSelected += delegate
{

```

```

Fast_Market.VistaCliente
OnCreate(Bundle savedInstanceState)

vprecios.Text = conex.TraerPrecio(listas.SelectedItem.ToString()).ToString();

});

agregar.Click += delegate
{
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    AlertDialog alert = dialog.Create();
    alert.SetTitle("Guardando Informacion!");
    alert.SetIcon(Android.Resource.Drawable.AlertLightFrame);
    alert.SetMessage("Esta Seguro de Guardar este Producto");
    alert.SetButton("No", (s, ev) =>
    {
        vprecios.Text = "";
    });
    alert.SetButton2("Si", (s, ev) =>
    {
        if (cantidad.Text == "" || vprecios.Text == "")
        {
            Toast.MakeText(this, "Ingrese todos los datos", ToastLength.Long).Show();
        }
    });
}

```

Linea: 208 Carácter: 1 SPC CRLF

```

Fast_Market.VistaCliente
OnCreate(Bundle savedInstanceState)

{
    if (cantidad.Text == "" || vprecios.Text == "")
    {
        Toast.MakeText(this, "Ingrese todos los datos", ToastLength.Long).Show();
    }
    else
    {
        productosactu.Clear();
        productosactu2.Clear();
        pedi = listas.SelectedItem.ToString() + " " + cantidad.Text;
        contar = contar + Convert.ToDouble(cantidad.Text) * Convert.ToDouble(vprecios.Text);
        lista2.Add(pedi);

        productos.Add(listas.SelectedItem.ToString());
        cantidades.Add(cantidad.Text);
        var adapter5 = new ArrayAdapter(this, Android.Resource.Layout.SimpleSpinnerItem, productos);
        listado.Adapter = adapter5;

        var adapter511 = new ArrayAdapter(this, Android.Resource.Layout.SimpleSpinnerItem, cantidades);
        listado2.Adapter = adapter511;

        totalp.Text = "$ " + contar.ToString() + " dolares";
        vprecios.Text = "";
    }
}

```

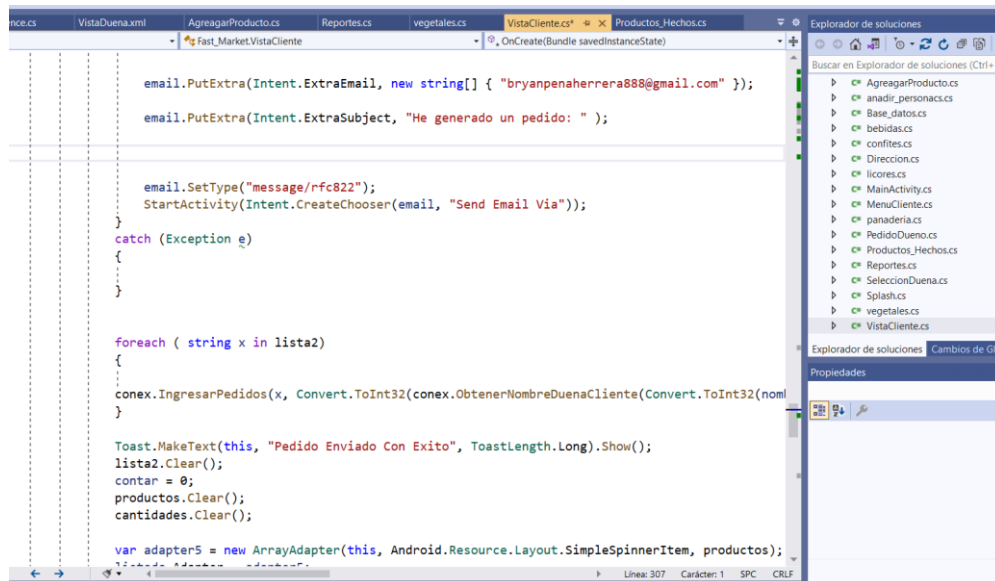
Linea: 208 Carácter: 1 SPC CRLF

```

enviar.Click += Button_Cliked;
enviar.Click += delegate
{
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    AlertDialog alert = dialog.Create();
    alert.SetTitle("Guardando Informacion!");
    alert.SetIcon(Android.Resource.Drawable.AlertLightFrame);
    alert.SetMessage("Esta Seguro de Enviar este Pedido");
    alert.SetButton("No", (s, ev) =>
    {
        });
    alert.SetButton2("Si", (s, ev) =>
    {
        try
        {
            Intent email = new Intent(Intent.ActionSendMultiple);

            email.PutExtra(Intent.ExtraEmail, new string[] { "bryanpenaherrera888@gmail.com" });

```



```

email.PutExtra(Intent.ExtraEmail, new string[] { "bryanpenaherrera888@gmail.com" });
email.PutExtra(Intent.ExtraSubject, "He generado un pedido: " );

email.SetType("message/rfc822");
StartActivity(Intent.CreateChooser(email, "Send Email Via"));
}
catch (Exception e)
{
}

foreach ( string x in lista2)
{
conex.IngresarPedidos(x, Convert.ToInt32(conex.ObtenerNombreDueniaCliente(Convert.ToInt32(noml
}

Toast.MakeText(this, "Pedido Enviado Con Exito", ToastLength.Long).Show();
lista2.Clear();
contar = 0;
productos.Clear();
cantidades.Clear();

var adapter5 = new ArrayAdapter(this, Android.Resource.Layout.SimpleSpinnerItem, productos);

```

```
1 referencia
public async void Button_Clicked(object sender, EventArgs e)
{
    var direccion = FindViewById<TextView>(Resource.Id.textView3);
    var nombrequi = Intent.GetStringExtra(Nombre_usuario);
    try
    {
        var location = await Geolocation.GetLastKnownLocationAsync();
        if (location != null)
        {
            location = await Geolocation.GetLocationAsync(new GeolocationRequest
            {
                DesiredAccuracy = GeolocationAccuracy.Medium,
                Timeout = TimeSpan.FromSeconds(30)
            });
            // Toast.MakeText(this, $"Latitude: {location.Latitude}, Longitude: {location.Longitude}, A
            if (location == null)
                Toast.MakeText(this, "No GPS", ToastLength.Short);
            else
                // direccion.Text = location.Latitude.ToString() + "," + location.Longitude.ToString();
                Toast.MakeText(this, "Envio de Direccion Exitosa", ToastLength.Short);
                conex.InsertarDireccion(Convert.ToInt32(nombrequi), location.Latitude.ToString() + "," + l
                Toast.MakeText(this, "Envio de Direccion Exitosa", ToastLength.Short);
        }
    }
}
```

3.3. Diseñar el circuito de lectura de tarjetas con interacción al sistema

Para la generación del circuito se debe tomar en cuenta los siguientes materiales que permitirán generar la lógica deseada.

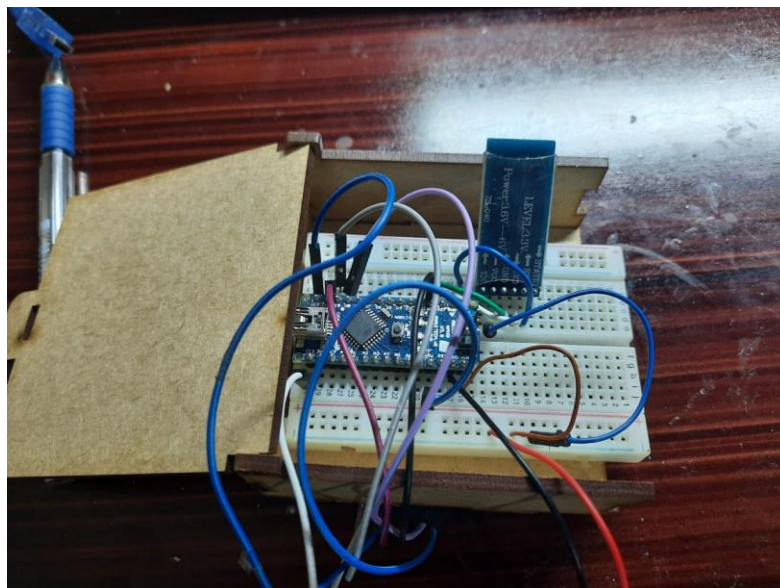
Materiales

- Arduino Nano
- Modulo RFID
- Tags
- Bluetooth hc-05
- Batería para la alimentación

Tabla de Conexiones

Módulo RC522	Arduino Uno, Nano
SDA (SS)	10
SCK	13
MOSI	11
MISO	12
IRQ	No conectado
GND	GND
RST	9
3.3V	3.3V

Creación del sistema de Rfid para el uso de tarjetas del personal de entregas





Codificación del Arduino para el funcionamiento del sistema

rfidtesis Arduino 1.8.15

Archivo Editar Programa Herramientas Ayuda

```
rfidtesis

#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522

MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522

void setup() {
  Serial.begin(9600); //Iniciamos La comunicacion serial
  SPI.begin(); //Iniciamos el Bus SPI
  mfrc522.PCD_Init(); // Iniciamos el MFRC522
  // Serial.println("Control de acceso:");
}

byte ActualUID[4]; //almacenará el código del Tag leído
byte Usuario1[4]= {0xA0, 0x17, 0xDF, 0x2B}; //código del usuario 1
byte Usuario2[4]= {0xC1, 0x2F, 0xD6, 0x0E}; //código del usuario 2
void loop() {
  // Revisamos si hay nuevas tarjetas presentes
  if ( mfrc522.PICC_IsNewCardPresent() )
  {
    //Seleccionamos una tarjeta
    if ( mfrc522.PICC_ReadCardSerial() )
    {
      // Enviamos serialmente su UID
      Serial.print(F("Card UID:"));
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        ActualUID[i]=mfrc522.uid.uidByte[i];
      }
      Serial.print(" ");
      //comparamos los UID para determinar si es uno de nuestros usuarios
    }
  }
}
```

rfidtesis Arduino 1.8.15

Archivo Editar Programa Herramientas Ayuda

```

rfidtesis
void loop() {
  // Revisamos si hay nuevas tarjetas presentes
  if ( mfrc522.PICC_IsNewCardPresent() )
  {
    //Seleccionamos una tarjeta
    if ( mfrc522.PICC_ReadCardSerial() )
    {
      // Enviamos serialmente su UID
      Serial.print(F("Card UID:"));
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        ActualUID[i]=mfrc522.uid.uidByte[i];
      }
      Serial.print(" ");
      //comparamos los UID para determinar si es uno de nuestros usuarios
      if(compareArray(ActualUID,Usuario1))
        Serial.println("Registrado Ashly Cabrera...");
      else if(compareArray(ActualUID,Usuario2))
        Serial.println("Acceso concedido...");
      else
        Serial.println("Acceso denegado no pertenece a la corporacion Fast...");

      // Terminamos la lectura de la tarjeta tarjeta actual
      mfrc522.PICC_HaltA();
    }
  }
}

//Función para comparar dos vectores
boolean compareArray(byte array1[],byte array2[])
{
  if(array1[0] != array2[0])return(false);
  if(array1[1] != array2[1])return(false);
  if(array1[2] != array2[2])return(false);
  if(array1[3] != array2[3])return(false);
  return(true);
}

```

rfidtesis Arduino 1.8.15

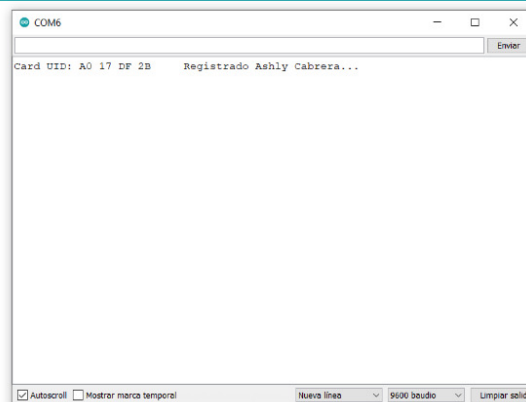
Archivo Editar Programa Herramientas Ayuda

```

byte ActualUID[4]; //almacenará el código del Tag leído
byte Usuario1[4]= {0xA0, 0x17, 0xDF, 0x2B}; //código del usuario 1
byte Usuario2[4]= {0xC1, 0x2F, 0xD6, 0x0E}; //código del usuario 2
void loop() {
  // Revisamos si hay nuevas tarjetas presentes
  if ( mfrc522.PICC_IsNewCardPresent() )
  {
    //Seleccionamos una tarjeta
    if ( mfrc522.PICC_ReadCardSerial() )
    {
      // Enviamos serialmente su UID
      Serial.print(F("Card UID:"));
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        ActualUID[i]=mfrc522.uid.uidByte[i];
      }
      Serial.print(" ");
      //comparamos los UID para determinar si es uno de nuestros usuarios
      if(compareArray(ActualUID,Usuario1))
        Serial.println("Registrado Ashly Cabrera...");
      else if(compareArray(ActualUID,Usuario2))
        Serial.println("Acceso concedido...");
      else
        Serial.println("Acceso denegado...");

      // Terminamos la lectura de la tarjeta tarjeta actual
      mfrc522.PICC_HaltA();
    }
  }
}

```



3.4. Diseño de una app de tipo móvil que integre tecnología RFID para el control de entregas del servicio a domicilio.

Para el control de los vendedores y el estado de cada uno de ellos se detalló los siguientes requerimientos que solventan lo solicitado.

CODIGO DEL REQUERIMIENTO	RF04
NOMBRE	El sistema tiene la capacidad de registrar y monitorear a los repartidores.
DESCRIPCION	El Sistema permite que el dueño de la tienda registre a sus repartidores y poder controlar quien está libre u ocupado.
ENTRADA	Nombres (alfanumérico) Apellidos (alfanumérico)
SALIDA	Repartidor ingresado.
PRIORIDAD	Alta

Código del Requerimientos	RF07
Nombre	El sistema tiene la capacidad sincronizarse con el sistema de RFID.
Descripción	El sistema obtiene la información del sistema de RFID a través del microcontrolador ingresado.
Entrada	Nombre Repartidor(carácter) Estado(carácter)
Salida	Recibir estado del repartidor.
Prioridad	Alta



Para poder acceder a esta vista o funcionalidad se debe entrar como el dueño del local y dar clic en personal



Posteriormente se accede a la vista en la cual cuenta con los empleados que trabajan para realizar las entregas.



Los cuales se pueden ver si se encuentran libres u ocupados en caso de estar libre pues se da clic en el botón ocupar para que envíe un mensaje o notificación al WhatsApp del personal que se especificó.



Cuando el personal llegue pasara su tarjeta en el módulo RFID que se construyó y automáticamente cambiara su estado en la vista mostrada anteriormente.

Código generado para las vistas y la extracción de la información a través del módulo RFID

```

VistaCliente.cs Personalfinal.cs* Personal.xml* VistaDuenaxml SeleccionDuenacs
Fast Market Fast_Market.Personalfinal OnCreate(Bundle save
19 {
20 [Activity(Label = "Personalfinal")]
  -referencias
21 public class Personalfinal : Activity
22 {
23     private Java.Lang.String dataToSend;
24     //Variables para el manejo del bluetooth Adaptador y Socket
25     private BluetoothAdapter mBluetoothAdapter = null;
26
27     private BluetoothSocket btSocket = null;
28     //Streams de lectura I/O
29     private Stream outputStream = null;
30     private Stream inputStream = null;
31     //MAC Address del dispositivo Bluetooth
32     string whatsapp = "com.whatsapp";
33     TextView estadopersonal1;
34     TextView estadopersonal2;
35     TextView estadopersonal3;
36     //Id Unico de comunicacion
37     private static UUID MY_UUID = UUID.FromString("00001101-0000-1000-8000-00805F9B34FB");
38     //Metodo de verificacion del sensor Bluetooth
39     private static BluetoothAdapter bluetoothAdapter = null;
40     private const int REQUEST_ENABLE_BT = 2;
41     public static ParcelUuid UUID1;
42     public static Dictionary<string, string> deviceDictionary = new Dictionary<string, string>();
43     public static int STATE = 0;
44     public static BluetoothSocket mySocket;
45     public static Activity _activity;
46     // private static BluetoothReceiver receiver;
  
```

```

ket Fast_Market.Personalfinal OnCreate(Bundle
public static Activity _activity;
// private static BluetoothReceiver receiver;
private static BluetoothDevice pairedBTDevice;
2 referencias
bool VerificarApp(String uri)
{
    try
    {
        Application.Context.PackageManager.GetPackageInfo(uri, PackageManager.PackageInfoFlags.of(PackageManager.PackageInfoFlags.GET_ACTIVITIES));
        return true;
    }
    catch (PackageManager.NameNotFoundException e)
    {
        return false;
    }
}
0 referencias
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);

    // Create your application here

    BluetoothAdapter blu = BluetoothAdapter.DefaultAdapter;
  
```

```
Personalfinal.cs*  Personal.xml*  VistaDuenaxml  SeleccionDuenaxml
Fast_Market.Personalfinal  * OnCreate(
BluetoothAdapter blu = BluetoothAdapter.DefaultAdapter;

base.OnCreate(savedInstanceState);
Xamarin.Essentials.Platform.Init(this, savedInstanceState);
// Set our view from the "main" layout resource
SetContentView(Resource.Layout.Personal);

TextView lista = FindViewById<TextView>(Resource.Id.textView10);
var ashly = FindViewById<Button>(Resource.Id.button1);
var bryan = FindViewById<Button>(Resource.Id.button2);

// var ventilacion = FindViewById<ToggleButton>(Resource.Id.toggleButton2);
// var puerta = FindViewById<ToggleButton>(Resource.Id.toggleButton3);
estadopersonal1 = FindViewById<TextView>(Resource.Id.textView11);
estadopersonal2 = FindViewById<TextView>(Resource.Id.textView13);
estadopersonal3 = FindViewById<TextView>(Resource.Id.textView14);
//luz.CheckedChange += tgConnect_HandleCheckedChange;

//si se activa el toggle button se inicial el metodo de conexion
// Connect();
var listOfDevices = blu.BondedDevices;

CheckBt();
Connect();
ashly.Click += delegate
```

```

Personalfinal.cs*  Personal.xml*  VistaDuena.xml  SeleccionDuena.cs
Fast_Market.Personalfinal  OnCreate(Bundle savedInstanceState)

CheckBt();
Connect();
ashly.Click += delegate
{
    // beginListenForData();

    estadopersonal1.Text = "Ocupado";

    if (VerificarApp(whatsapp))
    {
        Intent intent = new Intent();
        intent.SetAction(Intent.ActionSend);
        intent.PutExtra(Intent.ExtraText, "Confirmacion del pedido se lo entregara aproximadamente en min");
        intent.SetType("text/plain");
        intent.SetPackage(whatsapp);
        StartActivity(intent);
    }
    else
    {
        Toast.MakeText(this, "WhatsApp no está instalado. No se pudo enviar el mensaje", ToastLength.Long).Show();
    }
};

bryan.Click += delegate
{
    toast.MakeText(this, "WhatsApp no esta instalado. No se pudo enviar el mensaje", toastLength.Long).Show();
};

bryan.Click += delegate
{
    // beginListenForData();

    estadopersonal2.Text = "Ocupado";
    if (VerificarApp(whatsapp))
    {
        Intent intent = new Intent();
        intent.SetAction(Intent.ActionSend);
        intent.PutExtra(Intent.ExtraText, "Confirmacion del pedido se lo entregara aproximadamente en min");
        intent.SetType("text/plain");
        intent.SetPackage(whatsapp);
        StartActivity(intent);
    }
    else
    {
        Toast.MakeText(this, "WhatsApp no está instalado. No se pudo enviar el mensaje", ToastLength.Long).Show();
    }
};
}

```

```
Fast_Market.Personalfinal
/////////////////////////////////COneccion blu
///
1 referencia
private void CheckBt()
{
    //asignamos el sensor bluetooth con el que vamos a trabajar
    mBluetoothAdapter = BluetoothAdapter.DefaultAdapter;

    //Verificamos que este habilitado
    if (!mBluetoothAdapter.Enable())
    {
        Toast.MakeText(this, "Bluetooth Desactivado",
            ToastLength.Short).Show();
    }
    //verificamos que no sea nulo el sensor
    if (mBluetoothAdapter == null)
    {
        Toast.MakeText(this,
            "Bluetooth No Existe o esta Ocupado", ToastLength.Short)
            .Show();
    }
}
```

```

1 referencia
public void Connect()
{
    //TextView lista = FindViewById<TextView>(Resource.Id.textView10);

//
    string address = lista.Text;
    //Iniciamos la conexión con el arduino
    BluetoothDevice device = mBluetoothAdapter.GetRemoteDevice("98:D3:31:FC:68:43");
    System.Console.WriteLine("Conexión en curso" + device);

    //Indicamos al adaptador que ya no sea visible
    mBluetoothAdapter.CancelDiscovery();
    try
    {
        //Iniciamos el socket de comunicación con el arduino
        btSocket = device.CreateRfcommSocketToServiceRecord(MY_UUID);
        //Conectamos el socket
        btSocket.Connect();
        Toast.MakeText(this, "si se pudo", ToastLength.Long).Show();
        System.Console.WriteLine("Conexión Correcta");
    }
    catch (System.Exception e)
    {
        //en caso de generarnos error cerramos el socket
        System.Console.WriteLine(e.Message);
        try
        {
            btSocket.Close();
        }
    }
}

4 referencias
public void beginListenForData()
{
    //Extraemos el stream de entrada
    try
    {
        inStream = btSocket.InputStream;
    }
    catch (System.IO.IOException ex)
    {
        System.Console.WriteLine(ex.Message);
    }
    //Creamos un hilo que estará corriendo en background el cual verificará si hay algún dato
    //por parte del arduino
    Task.Factory.StartNew(() =>
    {
        //declaramos el buffer donde guardaremos la lectura
        byte[] buffer = new byte[1024];
        //declaramos el número de bytes recibidos
        int bytes;
        while (true)
        {
            try
            {
                //leemos el buffer de entrada y asignamos la cantidad de bytes entrantes
                bytes = inStream.Read(buffer, 0, buffer.Length);
                //Verificamos que los bytes contengan información
                if (bytes > 0)
            }
        }
    });
}

```

```

Fast_Market.Personalfinal OnCreate(Bundle savedInstanceState)
}
//Creamos un hilo que estara corriendo en background el cual verificara si hay algun dato
//por parte del arduino
Task.Factory.StartNew(() =>
{
    //declaramos el buffer donde guardaremos la lectura
    byte[] buffer = new byte[1024];
    //declaramos el numero de bytes recibidos
    int bytes;
    while (true)
    {
        try
        {
            //leemos el buffer de entrada y asignamos la cantidad de bytes entrantes
            bytes = inStream.Read(buffer, 0, buffer.Length);
            //Verificamos que los bytes contengan informacion
            if (bytes > 0)
            {
                //Corremos en la interfaz principal
                RunOnUiThread(() =>
                {
                    //Convertimos el valor de la informacion llegada a string
                    string valor = System.Text.Encoding.ASCII.GetString(buffer);
                    //Agregamos a nuestro label la informacion llegada
                    if (valor == "Ashly Cabrera")
                    {
                        estadopersonal1.Text = "Libre";
                    }
                    if (valor == "Brvan Penaherrera")
                }
            }
        }
    }
}

```

Código del Arduino que permite enviar la información para que se reciba en el celular mediante bluetooth.

```

sketch_dec12a Arduino 1.8.15
Archivo Editar Programa Herramientas Ayuda
sketch_dec12a$
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522

MFRC522 mfrc522(SS_PIN, RST_PIN); //Creamos el objeto para el RC522

void setup() {
    Serial.begin(9600); //Iniciamos La comunicacion serial
    SPI.begin(); //Iniciamos el Bus SPI
    mfrc522.PCD_Init(); // Iniciamos el MFRC522
}

byte ActualUID[4]; //almacenará el código del Tag leído
byte Usuario1[4]= {0xA0, 0x17, 0xDF, 0x2B}; //código del usuario 1
byte Usuario2[4]= {0xC1, 0x2F, 0xD6, 0x0E}; //código del usuario 2
void loop() {
    String data = Serial.readStringUntil('\n');
    Serial.println(data);
    // Revisamos si hay nuevas tarjetas presentes
    if ( mfrc522.PICC_IsNewCardPresent() )
    {
        //Seleccionamos una tarjeta
        if ( mfrc522.PICC_ReadCardSerial() )
        {
            // Enviamos serialmente su UID
            Serial.print(F("Card UID:"));
            for (byte i = 0; i < mfrc522.uid.size; i++) {
                Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                Serial.print(mfrc522.uid.uidByte[i], HEX);
                ActualUID[i]=mfrc522.uid.uidByte[i];
            }
            Serial.print(" ");
        }
    }
}

```

4. CONCLUSIONES

Se evidencia, con este trabajo, que la implementación con el sistema electrónico y la app móvil se vincula de forma exitosa logrando una comunicación eficiente al momento de enviar y recibir información.

El registro del stock de la tienda se registra de forma correcta, los productos se encuentran clasificados y con sus respectivas unidades logrando tener un control de su inventario.

El sistema permite al usuario ingresar sus pedidos logrando hacer que se generen de manera correcta y efectiva.

El circuito permite comunicarse con la app móvil, recibiendo y mandando información la cual es procesada internamente de forma exitosa.

El sistema cuenta con una interacción sincronizada del circuito llevando un control de los vendedores y de la generación de pedidos que realizan los usuarios.

La lectura generada por radiofrecuencia que se utiliza por los sistemas RFID, nos permite dependiendo, la capacidad de frecuencia que se utilice ya sea baja, media o alta modificar el rango o la distancia para su detección llegando hasta los 100m manejando una alta frecuencia.

Los tags activos, permiten emitir un código de identificación con mucha mayor potencia, al encontrarse alimentados y en consecuencia, el alcance de lectura es mucho mayor que el de los tags pasivos.

5. RECOMENDACIONES

Para el uso adecuado de la aplicación se debe utilizar celulares con un Android 6.0 en adelante.

Al realizar los distintos pedidos o consultas, se debe contar con una buena conexión de internet, para que el funcionamiento de la aplicación junto con la base de datos sea de forma eficiente.

Para generar el traspaso de información de forma bidireccional con la interacción del sistema electrónico, el celular tiene que estar conectado al bluetooth.

Para tener un mayor alcance de la lectura por parte de los repartidores se deber tener una configuración en alta frecuencia del sistema de radiofrecuencia que nos permita tener un mayor rango.

Para incorporar un mayor control y eficiencia del aplicativo respecto al control de la entrega y para su trayecto, se podría sincronizar con una señalización inteligente, que avisen al repartidor de la presencia de peatones o vehículos en las áreas de cruce.

6. BIBLIOGRAFIA:

- Echeverría Lolo (2021). *Se reduce las ventas de las empresas y de las personas naturales*. Quito. Recuperado de <https://criteriosdigital.com/category/datos/>.
- Chang, F, & Lozano, A. (2013). *Desarrollo e implementación de un Sistema para el Control e Inventario Continuo, Utilizando Tecnología RFID, para la Biblioteca de la UPS* (tesis de pregrado). Universidad Politécnica Salesiana, Ecuador.
- Domínguez Mateos, F., Paredes Velasco, M., & Santacruz Valencia, L. (2015). *Programación Multimedia y Dispositivos Móviles*. España: RA-MA Editorial.
- Millonaria caída en las ventas privadas estiman gremios por estado de excepción. (27 de abril de 2021). El Comercio. Recuperado:
<https://www.elcomercio.com/actualidad/negocios/millonaria-caida-ventas-privadas-excepcion.html>
- Arroyo, N. (2011). *Información en el móvil*. Barcelona: Editorial UOC.
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (2020). El 82,3% de MiPymes en el Ecuador utilizan Internet. Recuperado de <https://www.telecomunicaciones.gob.ec/el-823-de-mipymes-enel-ecuador-utilizan-internet/>
- Pulido Romero, E., Núñez Pérez, J. Á., & Escobar Domínguez, Ó. (2019). *Base de datos*. México: Grupo Editorial Patria.
- Pestano Herrera, J. M. (2018). *Microcontrolador STM32: programación y desarrollo*. Madrid: RA-MA Editorial.
- Pérez, J. (10 de 11 de 2004). *Google Académico*. Obtenido de https://www.academia.edu/17264316/Micro?from=cover_page

- Oracle. (1 de 05 de 2021). Obtenido de <https://www.oracle.com/mx/database/what-is-database/>
- Cantillo, C., Ruora, M., & Cantillo, A. (2012). *Tendencias actuales en el uso de dispositivos móviles en educación*. Washington: Department of Human Development, Education and Culture of the Organization of American States.
- Genwords. (10 de 05 de 2021). Obtenido de <https://www.genwords.com/blog/evolucion-comercio-electronico>
- Teletronica. (21 de 05 de 2021). Obtenido de <https://telectronica.com/cual-es-el-origen-de-la-tecnologia-rfid/>
- Sietillo99. (21 de 05 de 2021). Obtenido de <https://sites.google.com/site/sietillo99/wimax-2>
- Espacio Honduras. (20 de 05 de 2021). Obtenido de <https://www.espaciohonduras.net/microsoft-visual-studio-concepto-y-que-es-y-para-que-sirve-microsoft-visual-studio>
- Digital Guide Ionos. (20 de 05 de 2021). Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>
- Universidad Internacional de Valencia. (12 de 10 de 2017). Obtenido de https://www.universidadviu.com/pe/actualidad/nuestros-expertos/rfid-que-es-y-como-funciona.////_Articulo_RFID/////
- <https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/5321/IIT-05-035A.pdf?sequence=1&isAllowed=y>
- https://scholar.google.es/scholar?hl=es&as_sdt=0%2C5&q=tecnolog%C3%ADa+rfid+radio+frequency+identification&btnG=&oq=tecnologia+rfid

- Adell, J. (1998): "Redes y educación", en J. de Pablos y J. Jiménez (coords.): Nuevas Tecnologías, Comunicación Audiovisual y Educación. Barcelona: CEDECS, pp. 177-212. Agre, Philip (1998): "The Distances of Education: Defining the Role of Information Technology in the University," *Academe*, Septiembre.