



FACULTAD DE CIENCIAS EMPRESARIALES
CARRERA PROFESIONAL DE INGENIERÍA DE SISTEMAS
DE INFORMACIÓN Y GESTIÓN

“IMPLEMENTACIÓN DE UN APP MOVIL EN LA
GESTIÓN DE PEDIDOS”

Trabajo de investigación para optar el grado académico de
Bachiller en Ingeniería de Sistemas de Información y Gestión

Presentado por:

Juan Carlos Córdova Naventa
Camilo Galindo Vidal

Lima – Perú

2019

ANEXO 6 CIENTÍFICA
RESULTADO DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN
PARA OPTAR EL GRADO DE BACHILLER

Fecha: ___21___/___05___/2019

Facultad	Ciencias Empresariales		
Carreras de pre grado			
Ingeniería de Sistemas Empresariales	<input type="checkbox"/>	Ingeniería de Sistemas de Información y Gestión - CPE	<input checked="" type="checkbox"/>
Ingeniería Económica y de Negocios	<input type="checkbox"/>	Administración de Empresas – CPE	<input type="checkbox"/>
Administración de Negocios Internacionales	<input type="checkbox"/>		
Administración de Empresas	<input type="checkbox"/>		
Marketing y Administración	<input type="checkbox"/>		
Código	Alumno	Correo electrónico	Teléfono
4200720031	Córdova Naventa, Juan Carlos	4200720031@ucientifica.edu.pe	957255528
100023853	Galindo Vidal, Camilo	cgalindov@cientifica.edu.pe	987618124

Título del trabajo de investigación:

IMPLEMENTACIÓN DE UN APP MÓVIL EN LA GESTIÓN DE PEDIDOS

El Jurado revisor concluye que el **Trabajo de Investigación** para optar el Grado de Bachiller es:

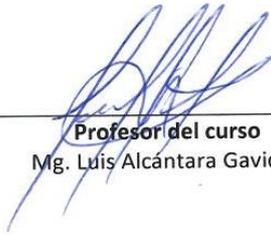
Aprobado (13 – 15)	<input checked="" type="checkbox"/>
Aprobado – Muy Buena (16 – 18)	<input type="checkbox"/>
Aprobado – Sobresaliente (19 – 20)	<input type="checkbox"/>
Desaprobado (0 – 12)	<input type="checkbox"/>

Calificándola con nota de: _14_ en letras (CATORCE _____)

Comentario/sugerencias/recomendaciones/observaciones:

-

Atentamente,



Profesor del curso
Mg. Luis Alcantara Gavidia

*Este documento puede ser enviado por correo electrónico

T // (51) 610 6400
opción 2
www.cientifica.edu.pe

Campus
Panamericana Sur
km 19 - Lima 42

Índice

1.1.	Planteamiento del Problema	6
1.2.	Antecedentes.	7
	1.2.1. Antecedentes Internacionales	7
	1.2.2. Antecedentes Nacionales	8
1.3.	Descripción del Problema	9
	1.3.1. Variables	12
	1.3.1.1. Variable independiente	12
	1.3.1.2. Variable dependiente	12
1.4.	Objetivo	13
	1.1.4. Objetivo General.	13
	1.4.2. Objetivos Específicos.	13
1.5.	Justificación	13
1.6.	Marco teórico	14
	1.6.1. Aplicación Móvil	14
	1.6.2. Dispositivos Móviles.	14
	1.6.3. Aplicación Móvil SIGESPED	15
	1.6.4. Implementación de una aplicación	15
	1.6.5. Cliente	16

1.6.6. Calidad	16
1.6.7. Servicio	16
1.6.8. Calidad del servicio	17
1.6.9. Satisfacción	17
1.7. Descripción del proyecto	17
1.7.1. Descripción de la solución de negocios propuesta	17
1.7.2. Arquitectura de procesos y de datos	19
1.7.3. Arquitectura de Sistemas	27
1.7.4. Arquitectura de la Infraestructura Tecnológica	29
1.8. Cronograma del proyecto	30
1.9. Presupuesto del proyecto	31
1.10. Indicadores de resultado del proyecto	32
1.10.1. Implementación de app SIGESPED	32
1.10.1.1. Puesta en producción de la aplicación y de los servicios web	33
1.10.1.1.1. Código Fuente de la actividad principal de consulta de menú en Android Studio	33

1.10.1.1.2. Código fuente del servicio web en PHP “consultaMenu.php”	39
1.10.1.2. Cantidad de pedidos exitosos usando SIGESPED	40
1.10.2. Tráfico de usuarios Delicass local Dasso en los meses de febrero y marzo	43
1.10.3. Simulación de costos en apertura de nuevos locales Delicass	49
Conclusiones	53
Referencias Bibliográficas	54

Índice de Tablas

Tabla 1: Locales Delicass	11
Tabla 2: Precios implementación Inforest	12
Tabla 3: Precios implementación Inforest	46

Índice de Figuras

Figura N° 1: Proceso durante el uso de la aplicación SIGESPED	19
Figura N° 2: Gestión de pedidos Antes	20
Figura N° 3: Gestión de pedidos – Después	21
Figura N° 4: Gestión de pedidos: usuario – Cliente Después.	22
Figura N° 5: Diagrama Entidad Relación. Acceso Seguridad	23
Figura N° 6: Diagrama Entidad Relación. Gestión de Pedidos	25
Figura N° 7: Diagrama Entidad Relación. Productos	26
Figura N° 8: Arquitectura de la Infraestructura Tecnológica	30

1.1. Planteamiento del problema

El Perú es un país que tiene una gran diversidad cultural y variada gastronomía donde cada ciudad tiene su propio plato típico con sabores inigualables, por lo que cuenta también con un gran número de restaurantes de diversos tamaños: grandes, pequeños y medianos. La tecnología se ha ido integrando cada vez más, debido a que es de vital importancia poder administrar y gestionar los recursos y la optimización de procesos en un restaurante como: pedidos, ingredientes, reservas, etc. Muñoz, 2017 (p 16) argumenta que los dispositivos móviles son una herramienta de uso frecuente para usuarios en restaurantes, y por tanto, los establecimientos se ven (y deberían verse) involucrarlos en sus procesos. Minchola, 2016 (p 16) nos da una idea de la importancia de la gestión de pedidos para la productividad de los restaurantes, así como generar una experiencia nueva para el cliente generando un buen servicio.

En Colombia se realizó un proyecto que plantea el desarrollo de un sistema integral para la gestión de pedidos en restaurantes (bares, cafeterías, etc.)

Suarez, Ramos, 2015 (p 4) nos refiere que la necesidad de realizar este proyecto tiene un factor de vital importancia que es el tiempo, ya que se encarga de gestionar de forma rápida la atención de los clientes, además de esta ventaja, quieren brindar al usuario una experiencia diferente de calidad y amigable.

En Perú se han desarrollado diferentes aplicaciones para la gestión de pedidos para restaurantes. Uno de ellos está enfocado en tres módulos, uno para el administrador de unidad de negocio, otro para el consumidor y otro para el pago, Muñoz, 2017 (p 18,19,20). La tendencia de llevar a cabo este tipo de proyectos va en aumento.

El restaurante gourmet Delicass es una empresa que cuenta con varios locales bien posicionados en sitios estratégicos, la mayoría de ellos en centros comerciales, los cuales cuentan con problemas en la gestión de pedidos cuando el local está a su máxima capacidad de aforo, esto genera malestar entre los clientes que deben esperar un poco más de tiempo para poder realizar su pedido.

1.2. Antecedentes.

En la última década se ha incrementado la implementación de sistemas web y aplicaciones móviles para mejorar y optimizar los procesos de los negocios, además de mejorar en la atención y la experiencia del usuario. En ese sentido, los restaurantes de nivel internacional, regional y nacional, también han implementado estos sistemas web y/o apps que ayuden a la reservación de mesas y/o toma de pedidos.

A nivel internacional podemos ver la experiencia de la empresa española ATECRESA, quienes desde 1982 se especializan en sistemas de gestión para restaurantes. Esta empresa el año 2013 lanzó su primera aplicación móvil para gestión de pedidos **MyOrder**, orientada a los clientes. La intención es que el cliente, si lo desea, pueda hacer su pedido sin necesidad de esperar al mozo; lo cual es muy oportuno en negocios en cuya hora punta tienen su aforo completo y los mozos no se dan abasto.

De esta forma, la aplicación no tiene intención de reemplazar al mozo sino de ayudar a una rápida atención de los clientes. Otro detalle a considerar es que la experiencia iniciada en Islas Canarias se ha expandido junto con la compañía a Inglaterra, Francia y Alemania. Otra experiencia internacional que podemos mencionar es la web de reservas Eltenedor.es, con presencia en más de 10000 restaurantes de España, Francia, Mónaco y Suiza. Si bien esta web solo permite hacer reservas, la cantidad de transacciones mensuales (400 000) nos da una idea de la importancia que tiene hoy en día para el cliente el ahorro de tiempo a la hora de darse un gusto en un restaurante.

En el ámbito regional latinoamericano, nos encontramos con el sistema colombiano **TouchBistro**, un sistema de punto de venta para iPad, que ayuda a los mozos en la toma de pedidos. De esta forma, al estar integrado con su sistema de gestión de restaurantes, los administradores de los mismos monitorean en tiempo real las operaciones de todos sus locales: cuánto se ha vendido, cuánto en efectivo, a crédito, etc. Sin embargo, la principal ventaja es que, al ser un punto de venta móvil, el mozo no pierde tiempo anotando en libreta y luego caminar hasta un punto de adición y registrando allí. Se busca y se obtiene la rapidez en la atención de pedidos.

A nivel nacional, en concreto en nuestra capital, tenemos como referencia el sistema **Restaurante.pe**, quien también es un completo sistema de gestión de

restaurantes con app Android para tabletas, destinada a la toma de pedidos por parte de los mozos. La ventaja de tener este tipo de punto de venta móvil ya la detalle líneas arriba: rapidez en la atención de los pedidos.

1.3. Descripción del Problema

En la actualidad, la empresa Delicass cuenta con 15 tiendas tanto en Lima como en provincias, de las cuales podemos apreciar en la tabla 1 la estimación de tiempo que demora la toma de pedidos. El tiempo estimado para la toma de pedidos es de 15 minutos, sin embargo, este tiempo puede aumentar hasta llegar a 20 minutos en hora punta. ¿A qué se debe esto? Primero, a que los mozos deben atender más mesas; segundo, a que los mozos deben acercarse al **punto de adición** para registrar el pedido. Se da frecuentemente el caso de que varios mozos hacen cola para ingresar sus pedidos en el punto de adición (se genera un cuello de botella), lo cual es indispensable porque si el pedido no se ingresa por el **punto de adición**, no llega la comanda de impresión a cocina (y no se procesa el pedido del cliente) ni a caja (para el cobro de la cuenta). Lo que muchas veces se dice es que la solución sería tener otro **punto de adición**, pero los costos para esto son muy elevados, tanto así que implementar un punto de adición extra cuesta como mínimo s/ 4800 soles

(ver la tabla 2).

Ante esta problemática, la solución no es tener más mozos (igual habría cuello de botella), sino tener uno o varios puntos de adición de bajo costo. Si a esto le agregamos que sea móvil, es decir, que el mozo no tenga que caminar para registrar

los pedidos, y que además el cliente mismo pueda generar su pedido sin la mediación del mozo, ayudaría a mayor fluidez, mayor satisfacción del cliente y ahorro significativo para la empresa. La solución es un aplicativo móvil disponible para dispositivos Android, que permita generar los pedidos ya sea a los mozos o los clientes directamente, reduciendo significativamente los costos por punto de adición, ya que cada dispositivo con el aplicativo instalado será un **punto de adición móvil**. Ante esta posible solución salta una dificultad: la empresa Delicass cuenta en todos los locales con el sistema Inforest, el cual es un sistema propietario. La solución a esta segunda dificultad es que la aplicación se comuniquen con la base de datos que utiliza el sistema Inforest por medio de web services, lo cual no afecta en nada al funcionamiento de Inforest, porque solo consultamos, escribimos y actualizamos la base de datos por medio de los

Webservices.

INFORMACION DE TIENDAS DELICASS					
N.º	TIENDA	N.º MESA	AFORO	TIEMPO ATENCION AL CLIENTE (PREPARACION) SANDWICHES y Ensaladas	POSTRES
1	AEROPUERTO	36	156	15 minutos	5 minutos
2	PLAZA NORTE	33	150	15 minutos	5 minutos
3	MAGDALENA	24	73	15 minutos	5 minutos
4	SALAVERRY	22	70	15 minutos	5 minutos
5	DASSO	31	152	15 minutos	5 minutos
6	MIRAFLORES	28	76	15 minutos	5 minutos
7	LARCOMAR	21	68	15 minutos	5 minutos
8	BEGONIAS	52	104	15 minutos	5 minutos
9	OPEN MALL	28	76	15 minutos	5 minutos
10	REAL PLAZA	13	40	15 minutos	5 minutos
11	JOCKEY PLAZA	29	66	15 minutos	5 minutos
12	POLO	25	53	15 minutos	5 minutos
13	CAMINOS	29	87	15 minutos	5 minutos
14	Real Plaza Cusco	28	76	15 minutos	5 minutos
15	Wanchap Cusco	21	68	15 minutos	5 minutos

Tabla 1: Locales Delicass

Descripción	Cantidad	Precio (dólares)	Dcto	Importe
Licencia Inforest Base	1.00	1,000.00	0.00	1,000.00
Licencia Inforest Punto de Venta	1.00	650.00	0.00	650.00
Valor de implementación	1.00	495.00	0.00	495.00

Tabla 2: Precios implementación Inforest

1.3.1. Variables

1.3.1.1. Variable independiente

Implementación de una aplicación móvil

1.3.1.2. Variable dependiente

Gestión de pedidos

1.4. OBJETIVOS

1.1.4. Objetivo General

- Implementar la aplicación móvil SIGESPED que mejore el proceso de gestión de pedidos en la empresa Delicass.

1.4.2. Objetivos Específicos

- Diseñar e implementar la aplicación móvil SIGESPED que se integre al proceso de gestión de pedidos en la empresa Delicass.
- Demostrar la influencia del uso de la aplicación móvil SIGESPED en el proceso de gestión de pedidos en la empresa Delicass.
- Demostrar la reducción de costos de implementación de nuevos puntos de adición con la implementación de la aplicación móvil SIGESPED en la empresa Delicass.

1.5. JUSTIFICACIÓN

Nuestra investigación sobre la implementación del aplicativo móvil se realiza por la necesidad de mejorar el proceso de Gestión de pedidos, renovándolo por medio de la tecnología móvil, pero además generando un ahorro considerable en costos a la empresa Delicass frente a su proceso tradicional. Este ahorro se verá reflejado no solo económicamente, sino también en tiempo de atención a los clientes, en control

de la actividad del restaurante y en la opción a futuro de escalar un sistema propietario, unificando las operaciones entre las sedes (actualmente cada sede cuenta con base de datos local).

Este estudio permitirá también mejorar la imagen de la empresa, brindando un aplicativo ágil y amigable con el cliente, mediante el cual el usuario podrá registrarse, lo cual permitirá a Delicass conocer los gustos de sus clientes. Esta opción permite a la aplicación ser aún más escalable, puesto que, a futuro, la empresa, conociendo los gustos de los usuarios podrá ofrecerles ofertas o descuentos de forma automatizada. Con la implementación de esta tecnología se demostrará cómo influye positivamente en el restaurante generando múltiples beneficios. Como agilizar el tiempo en la toma de pedidos, fidelizar más clientes, tener un registro de usuarios para ofrecer las promociones de forma automatizada, facilitar el modo de realización de reservas, aumentar el número de reservas realizadas, disminuir la cantidad de insumos perdidos.

1.6. Marco teórico

1.6.1. Aplicación Móvil

Es una aplicación informática especialmente diseñada para teléfonos inteligentes, de forma que solo puede ser ejecutada en estos dispositivos, específicamente en el sistema operativo para el cual fue diseñada. Ejemplo: una aplicación móvil diseñada para Android solo podrá ser ejecutada en un dispositivo con SO Android. Por lo general se encuentran disponibles a través de plataformas de distribución,

aplicaciones especiales que operan a manera de tienda de aplicaciones, operadas por las compañías propietarias de los sistemas operativos móviles (Morales, Guevara, Chavira & Álvarez, 2014).

1.6.2. Dispositivos Móviles

Un dispositivo móvil (DM) es el equipo que está libre de cableado. Su característica principal es su reducido tamaño, el cual favorece su movilidad. Estos dispositivos brindan grandes posibilidades de comunicación, lo que permite tener acceso a información y servicios independientemente de forma casi instantánea (Huelches García, Ronaldo Amador, Pineda Martínez, Ruano Sagastume, & Vanegas Barahona, 2014).

También se puede definir como un aparato de tamaño pequeño, con características como: capacidad de procesamiento, conexión permanente o intermitente a una red, memoria limitada, diseño específico para una función, pero que puede llevar a cabo otras funciones, uso individual (Guevara Soriano, 2010).

1.6.3. Aplicación Móvil SIGESPED

Es una aplicación móvil diseñada para ser ejecutada en tabletas con sistema operativo Android 4 o superior. El nombre SIGESPED significa Sistema de Gestión de Pedidos.

1.6.4. Implementación de una aplicación

El diseño y posterior desarrollo de una aplicación implica considerar desde la concepción de la idea hasta después de publicar el producto final. Durante este periodo de desarrollo del producto, diseñadores y desarrolladores trabajan de forma

coordinada (frecuentemente de forma simultánea). Las fases de este proceso se han resumido considerando únicamente la visión del diseño y el desarrollo del producto, es decir, no se tuvo en cuenta los roles de coordinación, la participación del cliente, ni los accionistas de la empresa (Cuello & Vittone, 2013).

El desarrollo de las aplicaciones para dispositivos móviles es muy diferente al desarrollo tradicional de productos de software, por lo que es normal que las metodologías usadas para estos entornos difieran de las usadas en el desarrollo de software clásico (Rahimian V. & Ramsin R., 2008).

1.6.5. Cliente

El cliente de un restaurante es aquella persona que solicita se le sirva un producto consumible preparado o distribuido por dicho restaurante. Kotler y Armstrong (2003) afirma que “el cliente es el centro de toda organización” Como todo ser humano que tiene necesidades y preocupaciones, pero sobre todo merece la mayor de las atenciones que la organización le pueda brindar.

1.6.6. Calidad

Según Norman Gaither (2000) “La calidad de un producto o servicio es el grado de percepción del cliente en que dicho bien cumple con sus expectativas”. Fernandez (2017) cita a Reeves y Bednar para afirmar que podemos entender la calidad como excelencia, como valor, como ajuste a las especificaciones del cliente y como satisfacción de las expectativas del cliente.

1.6.7. Servicio

Kotler & Armstrong (2003), afirma que un servicio es todo acto intangible, esencialmente intangible, que no necesariamente implica la propiedad de algo. Su creación no necesariamente está relacionada con un producto físico.

1.6.8. Calidad del servicio

“La calidad no es lo que se pone dentro de un servicio, es lo que el cliente obtiene de él y por lo que está dispuesto a pagar” (Drucker, 1990).

1.6.9. Satisfacción.

“La satisfacción es el nivel del estado de ánimo de una persona que resulta de comparar el rendimiento percibido de un producto o servicio con sus expectativas” (Kotler, 2006).

1.7. Descripción del proyecto

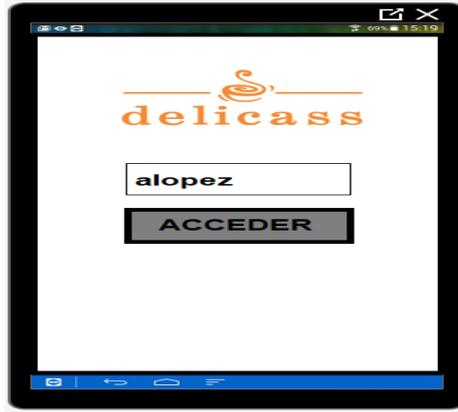
1.7.1. Descripción de la solución de negocios propuesta

1. Cada local de Delicass opera de manera independiente, **SIN ACCESO A UN SERVIDOR EN LA NUBE.**

2. En el local DASSO se utiliza la conexión WIFI, pero quien actúa como servidor es la PC usada en CAJA, es decir, es una red local.

3. El mozo usa el aplicativo, que ya tiene configurada la IP de la PC que actúa como servidor.

4. El mozo accede al aplicativo e inicia sesión



5. El mozo escoge una categoría



6. El mozo elige los productos y las cantidades según la categoría elegida



7. Una vez finalizado el pedido, este se visualiza en el sistema Inforest en la PC de caja. En paralelo se envía la impresión a la ticketera de cocina.
8. El cobro del pedido se hace desde caja.

1.7.2. Arquitectura de procesos y de datos

El sistema Inforest cuenta con una base de datos relacional SQL, manejada con el software SQL Server. La base de datos siempre es instalada en una PC con SO Windows, que actúa de servidor. En esta se instala además la caja y un punto de adición del sistema Inforest.

Vamos a utilizar esta PC con Windows, para instalar el software XAMPP, con Apache 2 y Php 7, de forma que por medio de código PHP se realicen los webs services, que a su vez harán las operaciones CRUD a la base de datos SQL.

Procesos durante el uso de la aplicación SIGESPED

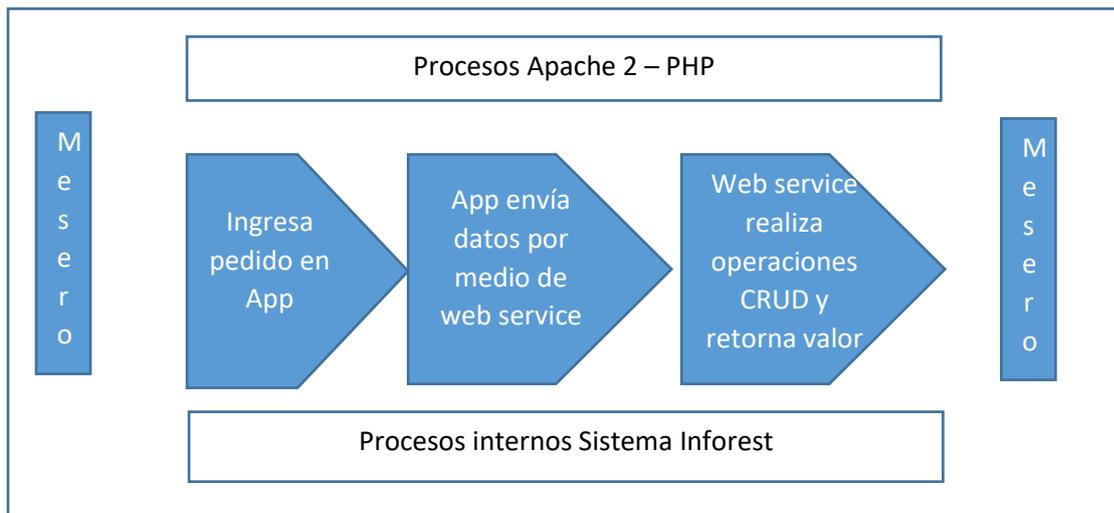


Figura N° 1: Arquitectura de procesos y de datos – Proceso durante el uso de la aplicación SIGESPED

Fuente: Elaboración propia.

Proceso de Gestión de pedidos: Antes

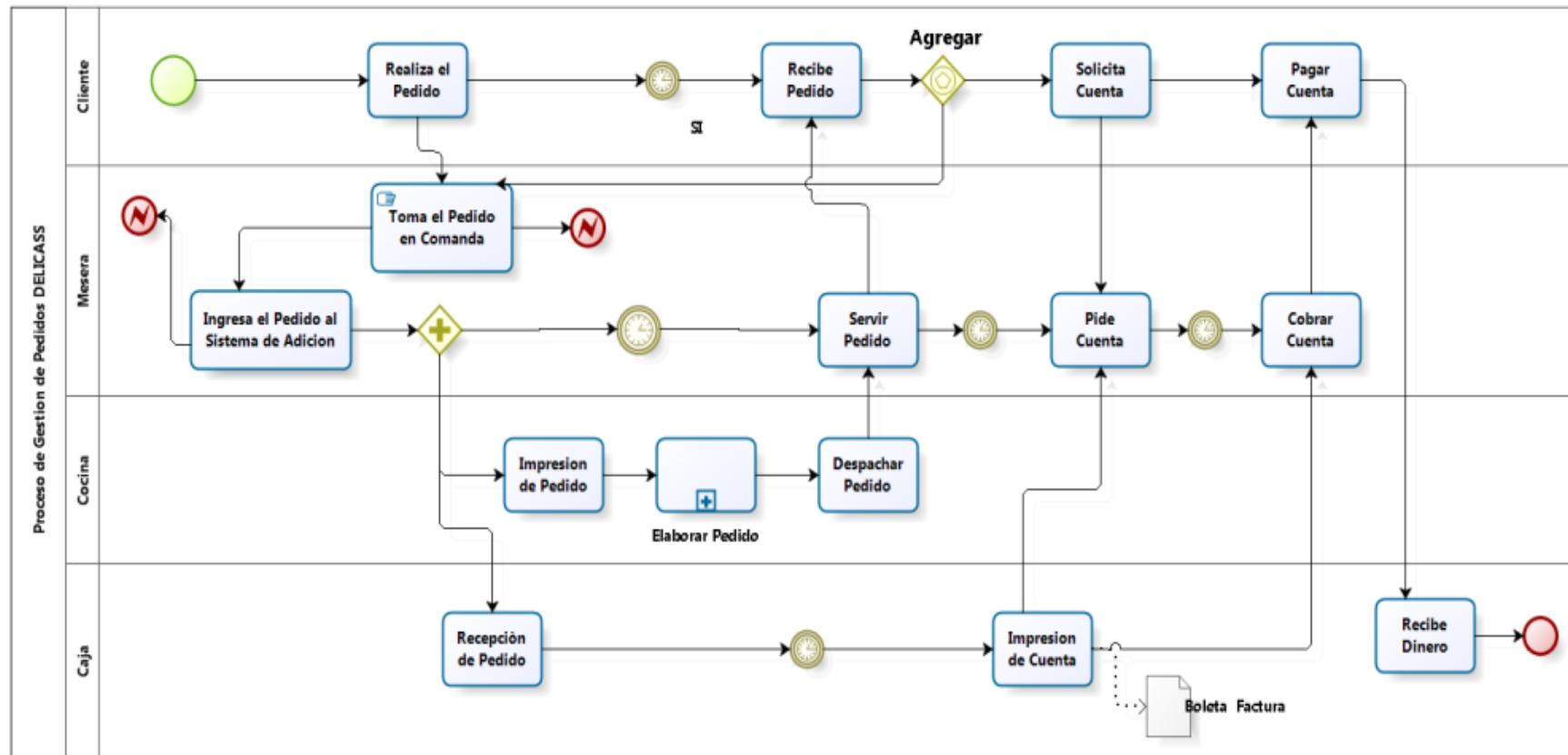


Figura N° 2: Arquitectura de procesos y de datos - Gestión de pedidos Antes

Fuente: Elaboración propia.

Proceso de Gestión de Pedidos: Después

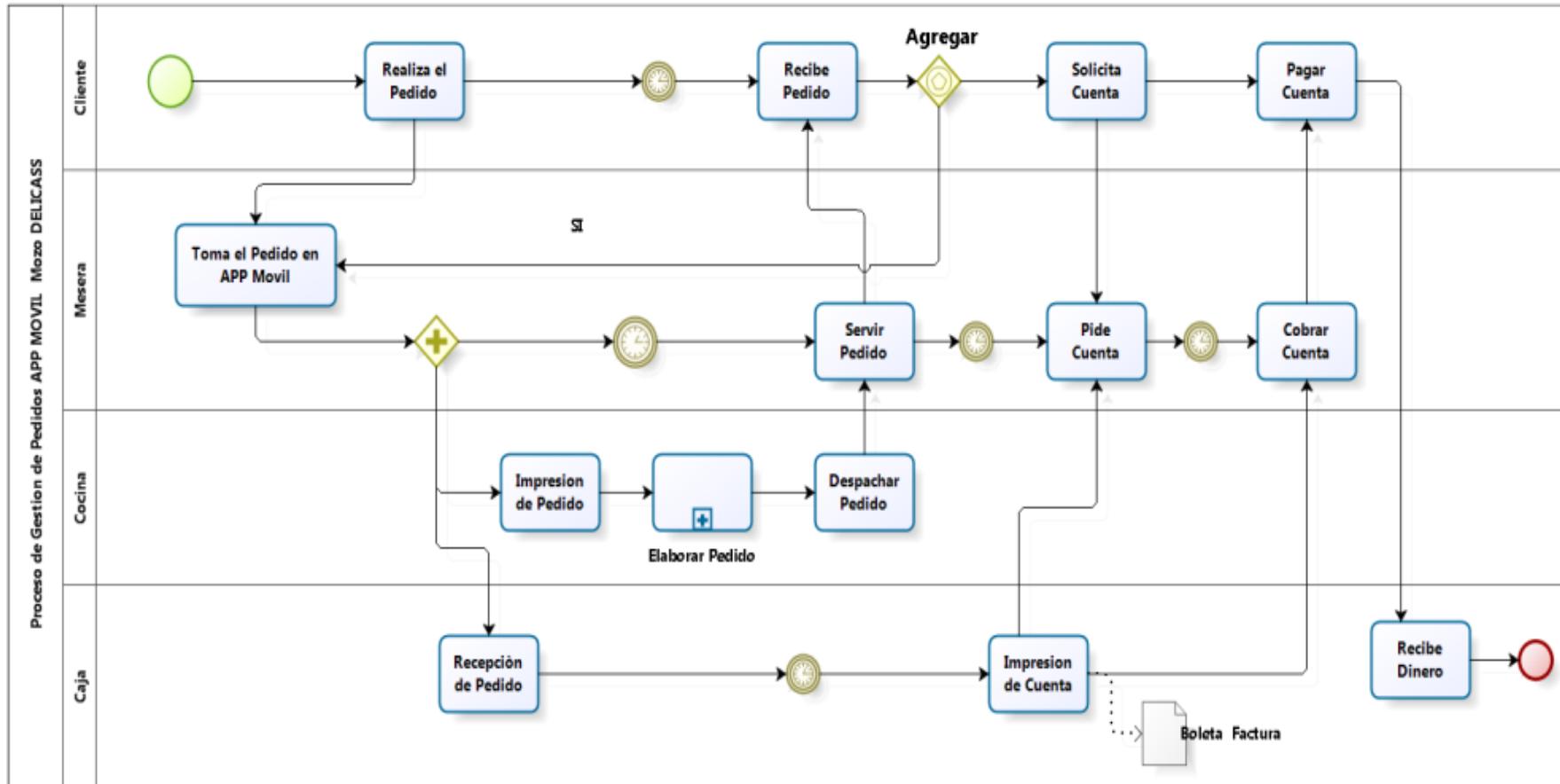


Figura N° 3: Arquitectura de procesos y de datos -Gestión de pedidos – Después

Fuente: Elaboración propia.

Proceso de Gestión de Pedidos Cliente: Después

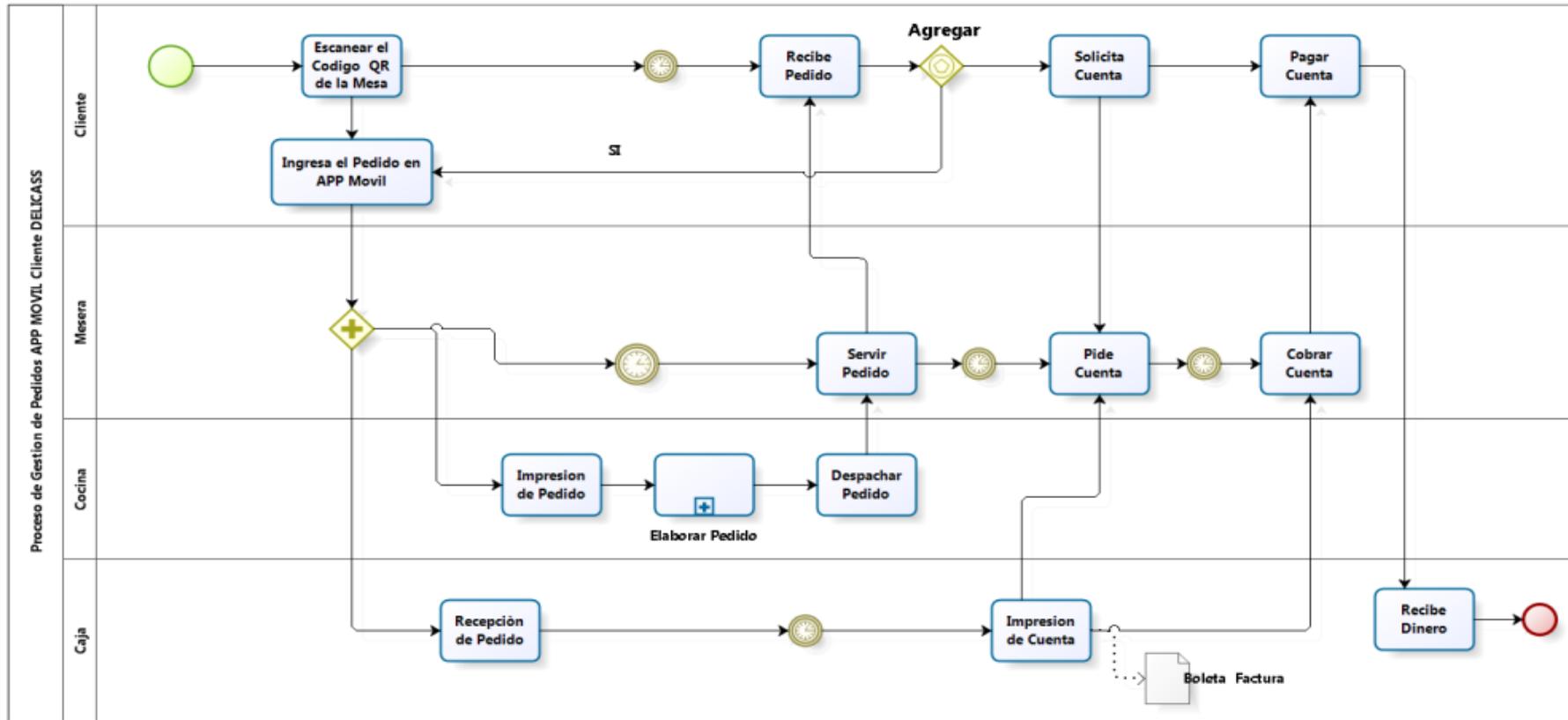


Figura N° 4: Arquitectura de procesos y de datos - Gestión de pedidos: usuario – Cliente Después.

Fuente: Elaboración propia.

Acceso - Seguridad

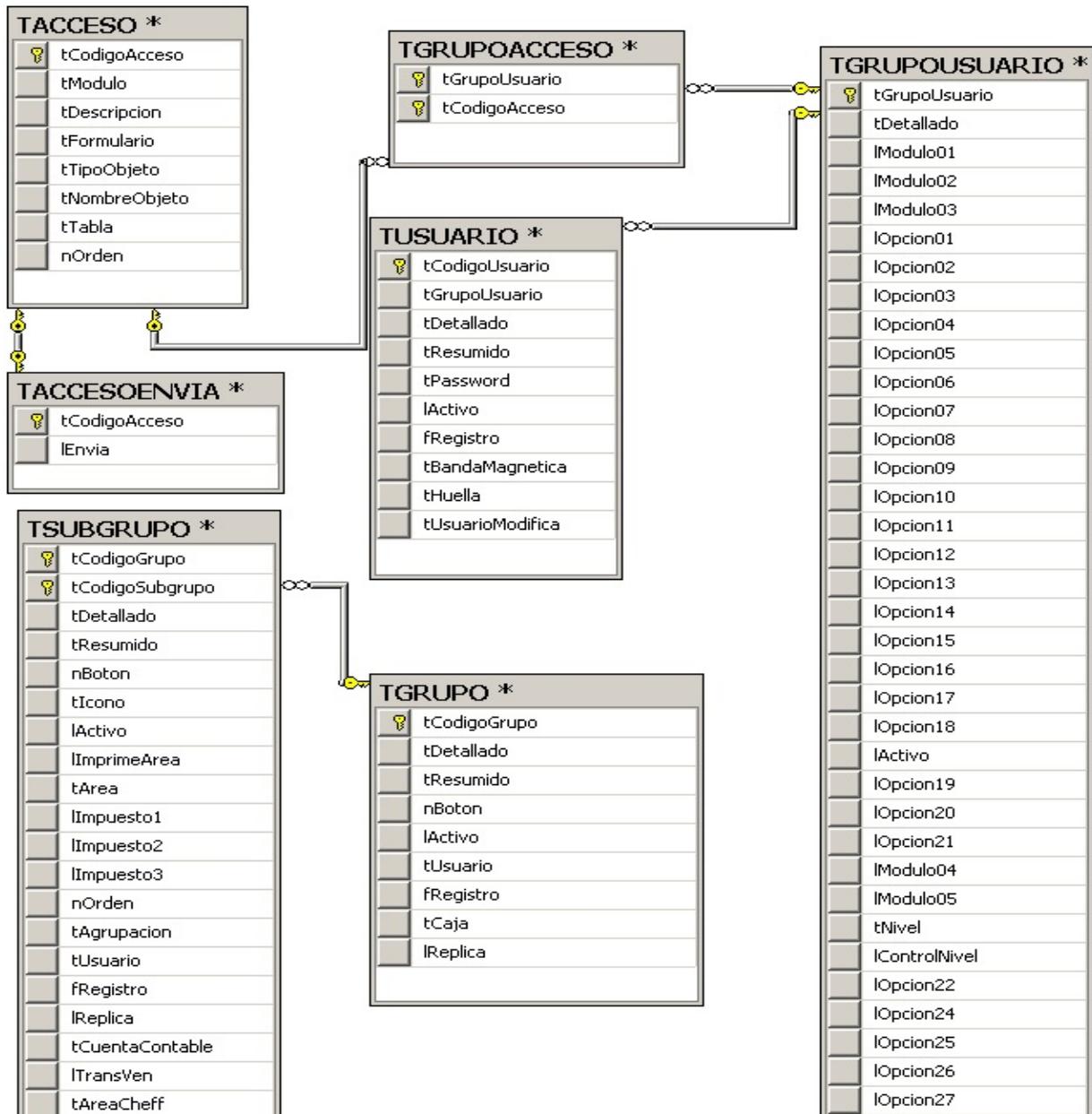


Figura N° 5: Arquitectura de procesos y de datos – Diagrama Entidad Relación. Acceso Seguridad

Fuente: Elaboración propia.

Características del Servidor de Datos:

- Sistema Operativo: Windows Server 2012 R2 Standard
- Procesador: Intel(R) CPU E3-1220 v5 @ 3.00 GHz
- Memoria instalada (RAM): 12.00 GB
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64
- Nombre de equipo: SERVER03
- Base de Datos: Microsoft SQL Server Enterprise Edición (64 bit)
- Sistema operativo: Microsoft Windows NT 6.1 (7601)
- Plataforma: NT x64
- Versión: 10.50 16001
- Procesadores: 04
- Intercalación del servidor: Modern Spanish CI AS

Gestion de Pedido

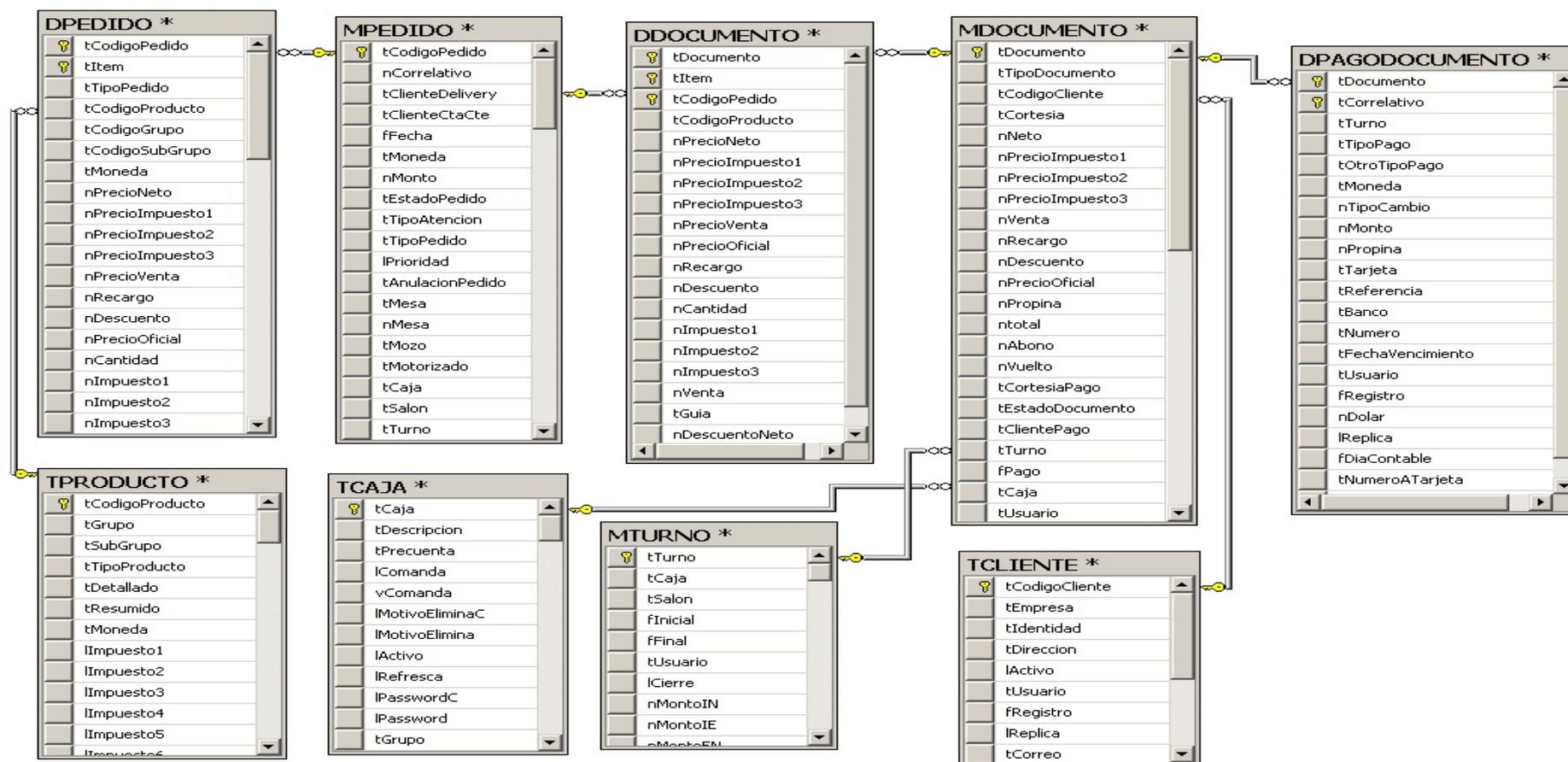


Figura N° 6: Arquitectura de procesos y de datos – Diagrama Entidad Relación. Gestión de Pedidos

Fuente: Elaboración propia.

Productos

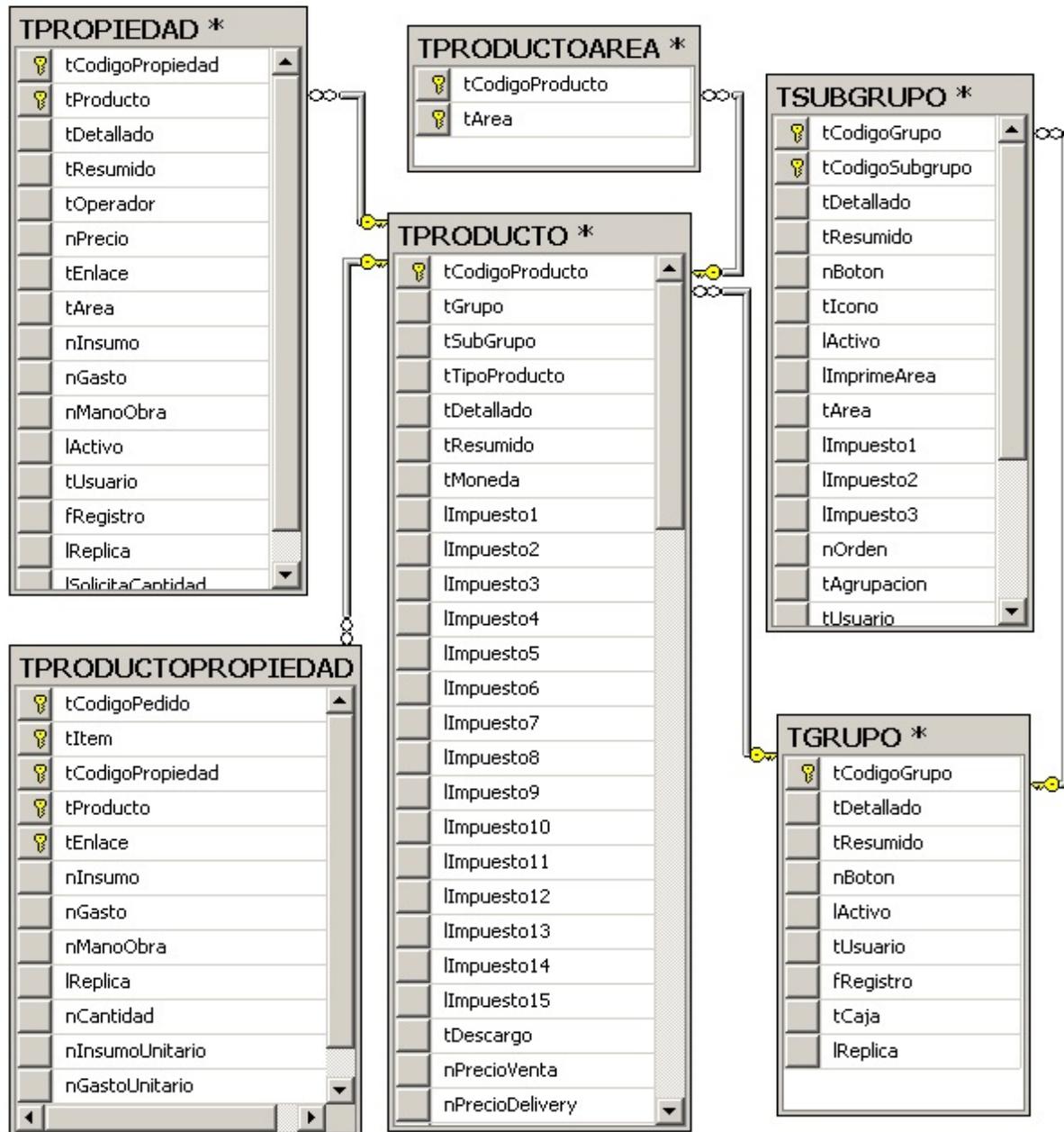


Figura N° 7: Arquitectura de procesos y de datos – Diagrama Entidad Relación. Productos

Fuente: Elaboración propia.

1.7.3. Arquitectura de Sistemas

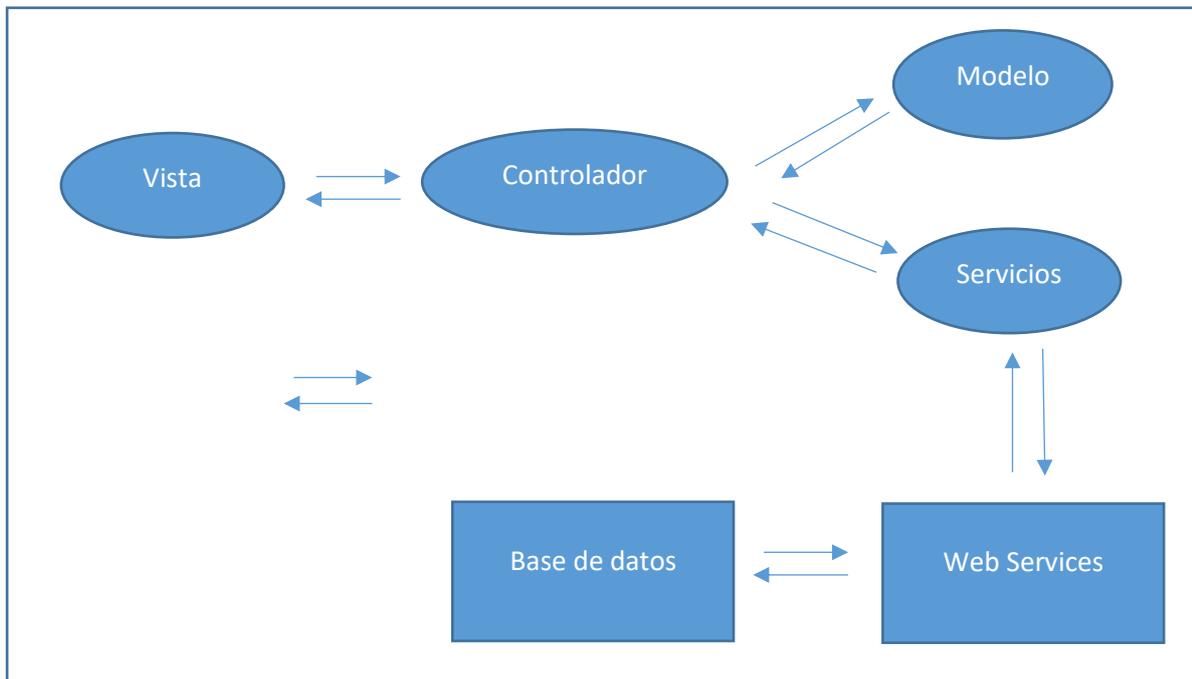
La aplicación **SIGESPED** (Sistema de gestión de pedidos online) es una aplicación desarrollada para el sistema operativo Android, con lenguaje de programación Java.

La aplicación tiene una arquitectura de 4 capas: Modelo – Vista – Controlador – Servicios.

1. En la capa de Modelo definimos tenemos las clases y sus métodos (bajo la lógica de la programación orientada a objetos). Estos métodos permiten escribir en memoria interna por medio de archivos de texto. Aquí se almacenarán básicamente datos del usuario.
2. En la capa de Vista manejamos las mismas por medio de archivos XML que se encuentran en la carpeta “layout” del proyecto.
3. En la capa de Servicios encontramos las clases que permiten la conexión con los webs services Php, por medio de peticiones Post. Una vez que el web services devuelve un valor, esta capa envía este valor a la capa de Controlador.
4. En la capa de Web services encontramos el código PHP responsable de hacer las operaciones CRUD en la base de datos. Esta capa envía una respuesta a la capa de servicios.

5. En la capa de Base de datos encontramos la base de datos del sistema Inforest.
Esta es una base de datos relacional SQL.

6. En la capa de Controlador encontramos las clases que manejan las vistas obteniendo los datos de la capa Modelo y de la capa Servicios. Finalmente, el usuario de la aplicación visualiza en pantalla los resultados.



1.7.4. Arquitectura de la Infraestructura Tecnológica

Conectividad: Cada sede debe tener conexión wifi para los dispositivos móviles. En esa misma red deben estar conectados el punto de adición y la caja, junto con las etiqueteras.

Dispositivos de entrada:

- Tablets o smartphones con la aplicación **SIGESPED** instalada
- Punto de adición

Dispositivos de salida:

- Etiqueteras de cocina y de caja
- Pantalla de caja

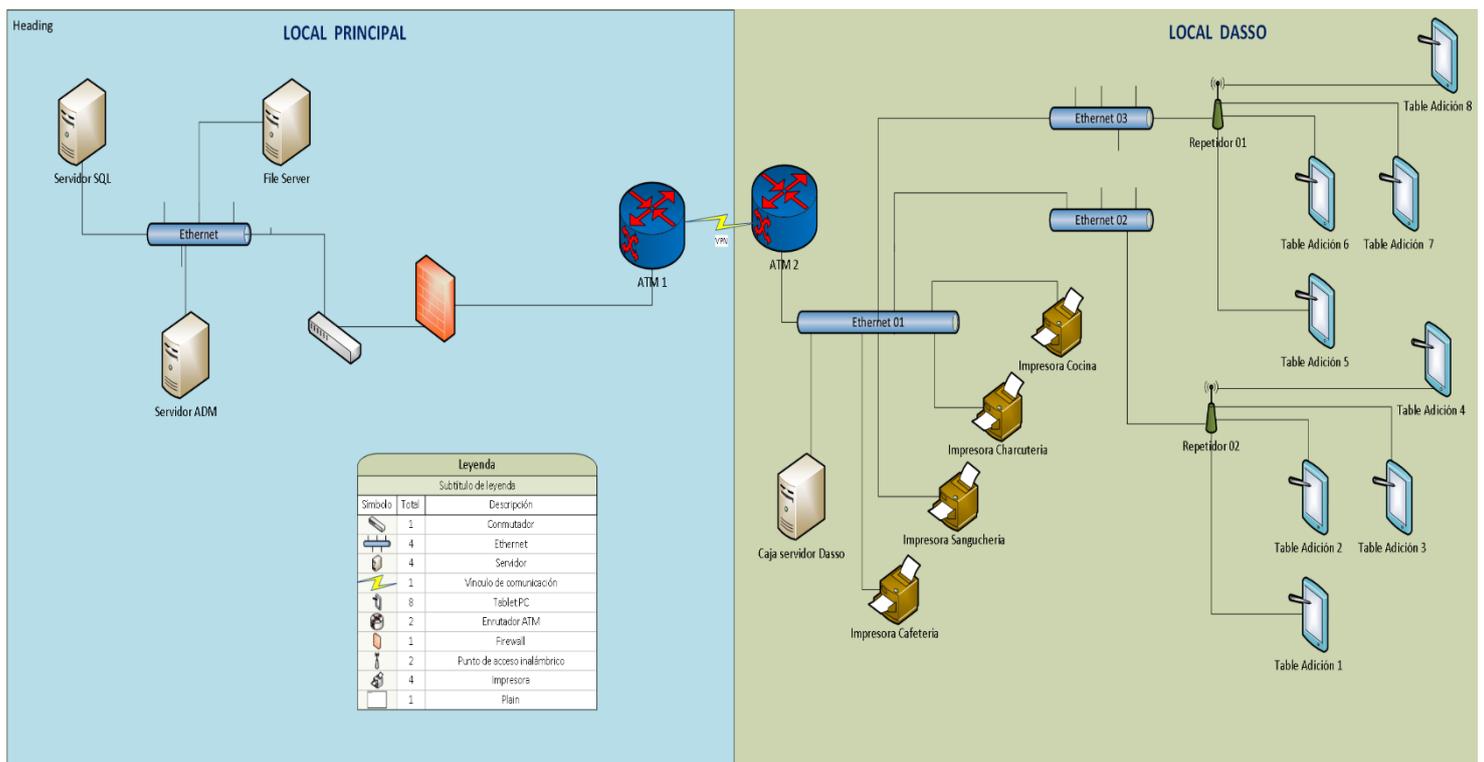


Figura N° 8: Arquitectura de la Infraestructura Tecnológica

Fuente: Elaboración propia.

1.8. Cronograma del proyecto

Cronograma de Actividades		APP SIGPOL PARA LA GESTIÓN DE PEDIDOS APLICADO A LA EMPRESA DELICASS							
Num	Tarea	Inicio	Final	enero-19	febrero-19	marzo-19	abril-19	mayo-19	
1	Conceptualización	3-1-19	11-1-19	■					
2	Definición	14-1-19	21-1-19	■					
3	Diseño	22-1-18	31-1-19	■					
4	Desarrollo	1-2-19	23-3-19		■	■			
5	Pruebas	25-3-19	28-3-19				■		
6	Publicación	29-3-19	31-3-19				■		

1.9. Presupuesto del proyecto

Nombre del Recurso	Cantidad	Precio por Hora	Horas de Trabajo Mensual	Mensual (Cantidad)	Costo Total
Costo de Desarrollo			480 horas	03 meses	
Diseñador de APP	01	S/. 30	45 horas		S/. 1,350
Desarrollador – Programador	01	S/. 30	300 horas		S/.9,000
QA-Tester	01	S/. 35	90 horas		S/. 3,150
Documentador de la APP	01	S/. 20	45 horas		S/. 900
Hardware: Laptop	02	S/. 3.5		3 meses	S/. 1,680
Servicio de Internet Wifi	01	S/. 1.25		3 meses	S/. 540
Total, Costo de Desarrollo					S/. 16,620

1.10. Indicadores de resultado del proyecto

En el presente trabajo se encuentran tres indicadores de resultado, los cuales se muestran en el siguiente cuadro:

Gestión de pedidos	KPI	Medición
Implementación App SIGESPED	% de avance e integración	Puesta en producción del app y los web service. Cantidad de pedidos exitosos usando SIGESPED
Tráfico de usuarios Delicass local Dasso en los meses de febrero y marzo	Valor de índice de correlación	Correlación entre tiempos involucrados en el proceso de Gestión de pedidos
Simulación de costos en apertura de nuevos locales Delicass	% de variación de costos	Comparación porcentual de costos entre la implementación Inforest y la implementación Inforest + SIGESPED

1.10.1. Implementación de app SIGESPED

Para llevar a cabo la implementación de la aplicación se tuvo que trabajar en dos entornos diferentes, pero complementarios. El primero de ellos es la aplicación en sí, desarrollada en Android Studio, mientras que el segundo entorno es el desarrollo de los Servicios web, desarrollados en lenguaje PHP, los cuales permitirán la conexión a la base de datos Inforest.

1.10.1.1. Puesta en producción de la aplicación y de los servicios web

Mostramos fragmentos del código fuente de la app Android y del service web.

1.10.1.1.1. Código Fuente de la actividad principal de consulta de menú en Android Studio

```
public class ConsultaMenu extends AppCompatActivity {
```

```
    String url1="", url2="", url3="";
    String datosi="";
    Button bt;
    String conectado=""; String estado=""; String conec3="";
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.prueba_socket);
```

```
    bt = (Button) findViewById(R.id.categoria);
    bt.setVisibility(View.GONE);
```

```
/*La variable url1 almacena la ruta general de acceso a la PC de caja del local
Dasso, la cual actuará como servidor. Se coloca :8080 porque es el puerto en el cual
configuramos el servidor Apache en ese equipo. La carpeta del proyecto es
"delicass" y el nombre del archivo que actúa como servicio web es
consultaMenu.php*/
```

```
    url1 = "http://192.168.43.62:8080/delicass";
    url2 = url1 + "/consultaMenu.php";
```

```
/*Conecta es una clase asíncrona que permite comunicación con el web service
mediante peticiones GET. Una vez recibida la respuesta, se envía al proceso
Lista()*/
```

```
    Conecta con = new Conecta();
    con.execute(url2);
```

```
    bt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```

        bt.setVisibility(View.GONE);
        Conecta con = new Conecta();
        con.execute(url2);
    }
});
}

```

```

public class Conecta extends AsyncTask<String,Void,String> {

```

```

    ProgressDialog progressDoalog;

```

```

    @Override

```

```

protected void onPreExecute() {
    progressDoalog = new ProgressDialog(EstablecerConexion.this);
    progressDoalog.setMax(100);
    progressDoalog.setMessage("Conectando....");
    progressDoalog.setTitle("");
    progressDoalog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressDoalog.show();
    super.onPreExecute();
}

```

```

    @Override

```

```

protected String doInBackground(String... params) {
    String url = getDatos(params[0]+"");
    return url;
}

```

```

    @Override

```

```

protected void onPostExecute(String result) {
    progressDoalog.dismiss();
    super.onPostExecute(result);
    conectado=result;
    Lista(conectado);
}

```

```

}

```

/*Los procesos Conecta2() y Conecta3() son al igual que Conecta() procesos asíncronos que permiten la conexión al servicio web respectivo. Lista() y Lista2() muestran al usuario los datos en pantalla acerca de los productos de las categorías del menú y de los productos de esa categoría respectivamente. Para eso usamos un elemento ListView*/

```

public class Conecta2 extends AsyncTask<String,Void,String> {

```

```

    ProgressDialog progressDoalog;

```

```

@Override
protected void onPreExecute() {
    progressDialog = new ProgressDialog(EstablecerConexion.this);
    progressDialog.setMax(100);
    progressDialog.setMessage("Conectando....");
    progressDialog.setTitle("");
    progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressDialog.show();
    super.onPreExecute();
}

@Override
protected String doInBackground(String... params) {
    String url = getDatos(params[0]+"");
    return url;
}

@Override
protected void onPostExecute(String result) {
    progressDialog.dismiss();
    super.onPostExecute(result);
    conectado=result;
    bt.setVisibility(View.VISIBLE);
    Lista2(conectado);
}
}

public class Conecta3 extends AsyncTask<String,Void,String> {

    ProgressDialog progressDialog;

    @Override
    protected void onPreExecute() {
        progressDialog = new ProgressDialog(EstablecerConexion.this);
        progressDialog.setMax(100);
        progressDialog.setMessage("Conectando....");
        progressDialog.setTitle("");
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.show();
        // TODO Auto-generated method stub
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... params) {
        // TODO Auto-generated method stub
        String url = getDatos(params[0]+"");
    }
}

```

```

        return url;
    }

    @Override
    protected void onPostExecute(String result) {
        // TODO Auto-generated method stub
        progressDialog.dismiss();
        super.onPostExecute(result);
        //conec3="";
        conectado=result;
    }
}

private void Lista(String mensaje){
    final ListView lista = (ListView) findViewById(R.id.mi_lista);
    String recepcionado[] = mensaje.split("#");
    int cont = recepcionado.length;

    datosi = mensaje;
    final String SP[] = new String[cont];
    final String tarray1[] = new String[cont];
    final String tarray2[] = new String[cont];
    final String tarray3[] = new String[cont];

    for(int i =0; i<cont; i++){
        String ndato[] = recepcionado[i].split("%");
        tarray1[i] = ndato[0];
        tarray2[i] = ndato[1];
        tarray3[i] = ndato[2];
        datosi = datosi + ndato[0] + " - " + ndato[1] + "\n";
    }

    //Toast.makeText(getApplicationContext(),datosi,Toast.LENGTH_LONG).show();
    AdapterLista2 adapterLista = new AdapterLista2(this,tarray1,tarray2,tarray3);
    lista.setAdapter(adapterLista);

    lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
        long id) {
            final int pos = position;
            url3 = url1 + "/consultaMenu2.php?dato="+tarray3[pos];
            Log.d("ip",url3);
            Conecta2 con = new Conecta2();

```

```

        con.execute(url3);

    }
});

}

```

```

private void Lista2(String mensaje){
    final ListView lista = (ListView) findViewById(R.id.mi_lista);
    String recepcionado[] = mensaje.split("#");
    int cont = recepcionado.length;

    datosi = mensaje;
    final String SP[] = new String[cont];
    final String tarray1[] = new String[cont];
    final String tarray2[] = new String[cont];
    final String tarray3[] = new String[cont];

    for(int i =0; i<cont; i++){
        String ndato[] = recepcionado[i].split("%");
        tarray1[i] = ndato[0];
        tarray2[i] = ndato[1];
        tarray3[i] = ndato[2];
        datosi = datosi + ndato[0] + " - " + ndato[1] + "\n";
    }

    //Toast.makeText(getApplicationContext(),datosi,Toast.LENGTH_LONG).show();
    AdapterLista2 adapterLista = new AdapterLista2(this,tarray1,tarray2,tarray3);
    lista.setAdapter(adapterLista);

    lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            final int pos = position;
            url3 = url1 + "/agregarProducto.php?dato="+tarray3[pos];
            Log.d("ip",url3);
            Conecta3 con = new Conecta3();
            con.execute(url3);

        }
    });
}

```

```

    }

    public String getDatos(String entrada) {
        URL alumUrl = null;
        String url2="";
        try{
            alumUrl = new URL(entrada);
            HttpURLConnection conn = (HttpURLConnection) alumUrl.openConnection();
            conn.connect();
            InputStream is = conn.getInputStream();
            byte[] array = new byte[1000];
            StringBuffer out = new StringBuffer();
            byte[] b = new byte[4096];
            for (int n; (n = is.read(b)) != -1;) {
                out.append(new String(b, 0, n, "UTF-8"));
            }
            String pot=new String(out.toString().getBytes("UTF-8"));
            url2=pot;
        }catch(IOException ex){
            ex.printStackTrace();
        }
        return url2;
    }
}

```

1.10.1.1.2. Código fuente del servicio web en PHP “consultaMenu.php”

```
<?php
/*Nombre del servidor al cual nos vamos a conectar*/
$dbserver = "DESKTOP-NLTLVMRT\dasso";
/*Usuario de la base de datos*/
$dbuname = "sa";
$dbpass = "";
$dbname = "master";
/*Librería que permite la conexión a SQL SERVER*/
include("adodb5/adodb.inc.php");

$db = ADONewConnection('mssql');
$db =& ADONewConnection('odbc_mssql');
$dsn = "Driver={SQL Server};
Server=DESKTOP-NLTLVMRT\dasso;
Database=INFORESTAEROP;";
$db->Connect($dsn,'sa','sistemas');

if(!$db->Connect($dsn,'sa','sistemas')){
}else {
}
$result = $db->Execute('SELECT * FROM dbo.TGRUPO');

$c = 0;
if ($result === false) die("failed");
while (!$result->EOF) {
    echo $result->fields[1].'% %'. $result->fields[0].'#';
    $result->MoveNext();
}
?>
```

Avance de implementación app SIGESPED:

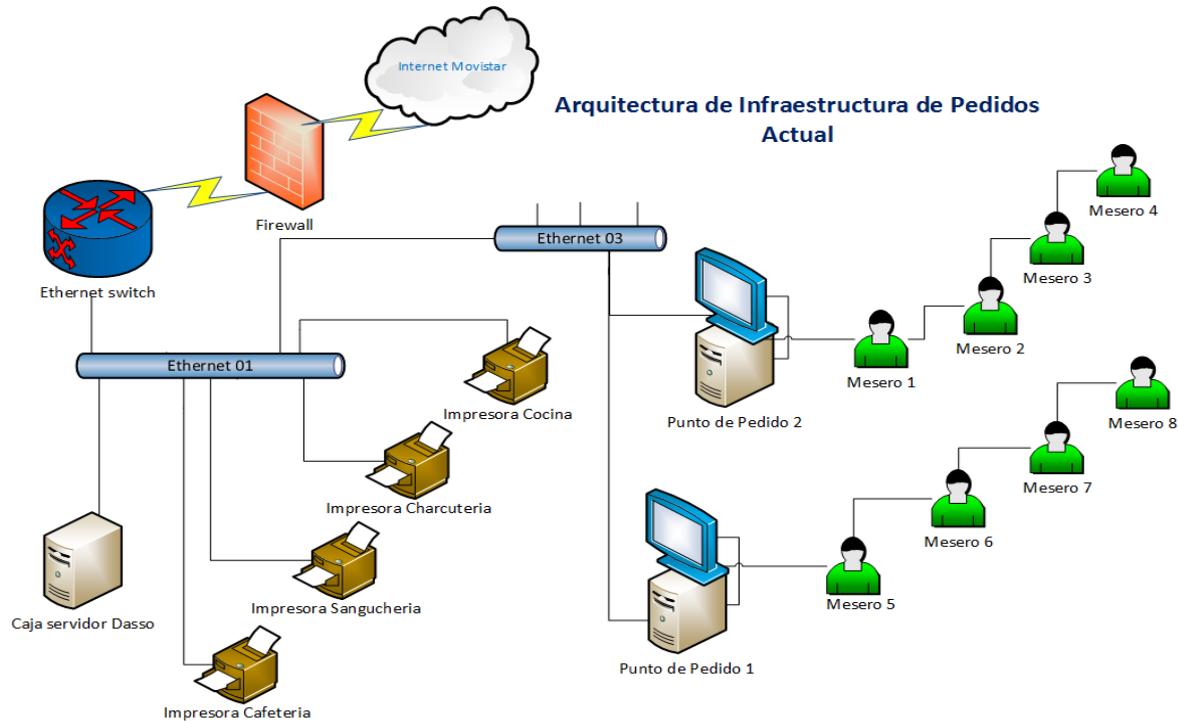
	CODIGO FUENTE	VISTAS
Aplicación SIGESPED	100%	90%
Servicios web	100%	No necesita

1.10.1.2. Cantidad de pedidos exitosos usando SIGESPED

En el documento adjunto “tabla_de_observaciones.xls” encontramos dos hojas con datos de observaciones propias de los meses de febrero y marzo. Encontraremos la cantidad de registros de boletas emitidas en el sistema Inforest en cada uno de esos meses.

Antes de implementar SIGESPED:

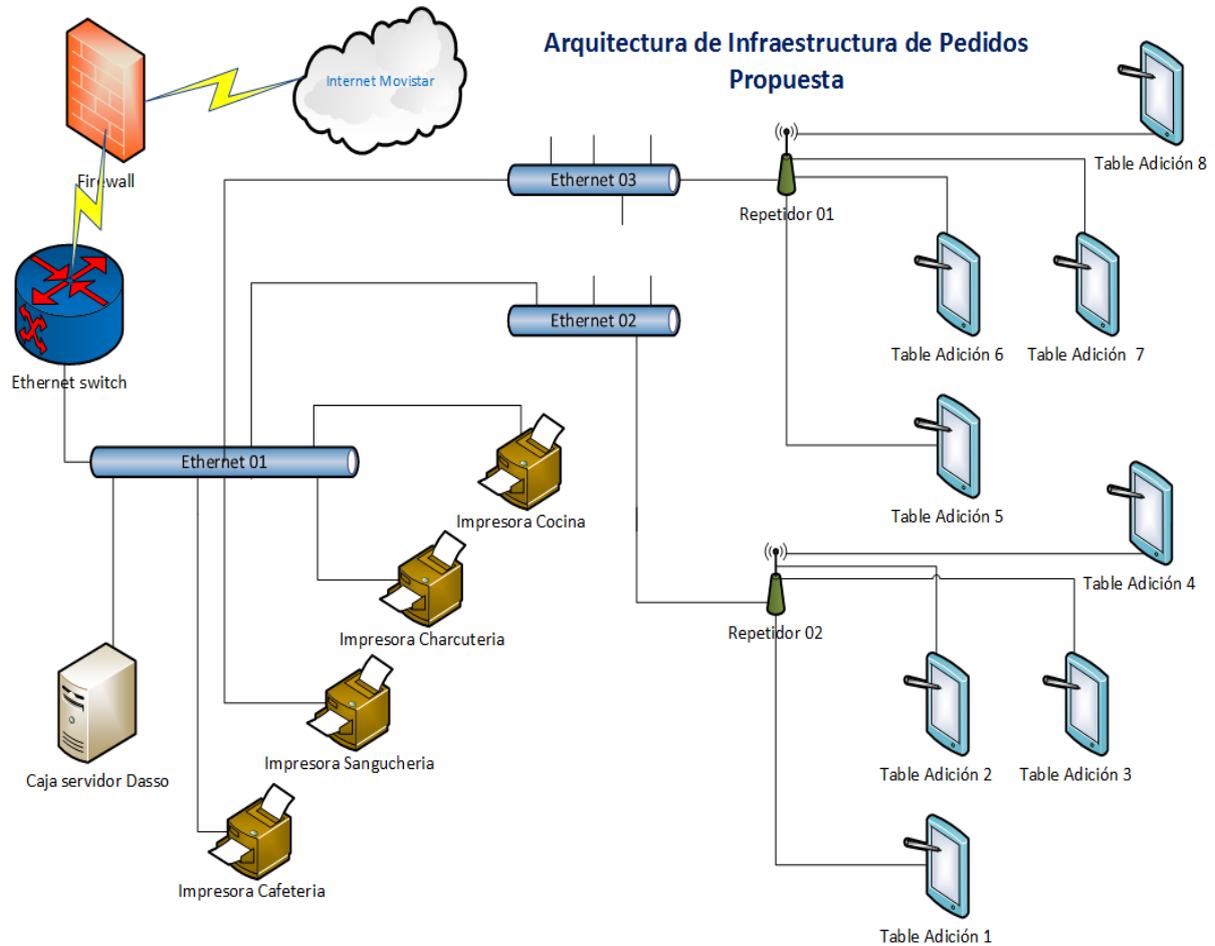
En el mes de febrero se emitieron 13820 boletas, únicamente usando el sistema Inforest. En este caso la distribución de los equipos es como sigue:



Arquitectura de Infraestructura Pedidos Actual		
Símbolo	Total	Descripción
	2	Ethemet
	3	Servidor
	2	Vínculo de comunicación
	1	Nube
	1	Enrutador ATM
	1	Firewall
	4	Impresora
	2	Monitor LCD

Luego de implementar SIGESPED:

En el mes de marzo se emitieron 12523 boletas, utilizando el sistema Inforest y la aplicación SIGESPED en paralelo. Las 12523 fueron emitidas por las tabletas con SIGESPED instalada, y registradas de manera adecuada por el sistema Inforest. Esto permite que luego el encargado de caja pueda visualizar los pedidos y realizar el cobro a los clientes. En este caso la distribución de equipos es como sigue:



Leyenda de Arquitectura de Infraestructura		
Propuesta		
Símbolo	Total	Descripción
	3	Ethernet
	1	Servidor
	2	Vínculo de comunicación
	1	Nube
	8	Tablet PC
	1	Enrutador ATM
	1	Firewall
	2	Punto de acceso inalámbrico
	4	Impresora

1.10.2. Tráfico de usuarios Delicass local Dasso en los meses de febrero y marzo

Los archivos adjuntos febrero.xlsx y marzo.xlsx contienen las observaciones propias acerca de los tiempos de permanencia de los clientes en el local Dasso.

Antes de implementar

A: Observaciones febrero en horario normal

Como muestra tenemos 20 mesas del local Dasso en horario normal.

Los datos observados fueron:

T1: hora en que el cliente se sienta en la mesa

T2: hora en que el cliente recibe la carta

T3: hora en que el cliente realiza su orden

T4: hora en que el mozo registra la orden en el punto de adición

De ello obtuvimos tres variables a observar:

A: Promedio de tiempo entre que el cliente se sienta y recibe la carta

B: Promedio de tiempo entre que el cliente recibe la carta y realiza su orden al mozo

C: Promedio de tiempo entre que el mozo recibe el pedido del cliente y lo ingresa en el punto de adición

* Todas las variables medidas en segundos

La información es la siguiente:

	A	B	C
MESA 1	102	272	180
MESA 2	108	308	192
MESA 3	120	320	210
MESA 4	96	260	180
MESA 5	108	278	186
MESA 6	126	338	210
MESA 7	108	296	204
MESA 8	120	332	216
MESA 9	108	284	192
MESA 10	102	266	180
MESA 11	114	308	204
MESA 12	126	332	216
MESA 13	96	266	190
MESA 14	102	278	180
MESA 15	108	296	198
MESA 16	114	296	198
MESA 17	120	320	205
MESA 18	120	330	220
MESA 19	114	300	192
MESA 20	108	305	200

Luego de analizar los datos por coeficiente de correlación obtuvimos los siguientes resultados:

Corr(A,B) 0.94316718
 Corr(B,C) 0.91991622
 Corr(A,C) 0.86206655

Esto indica que las variables están fuerte relacionadas en el horario observado.

B. Observaciones febrero en hora punta

Como muestra tenemos 20 mesas del local Dasso en hora punta.

Los datos observados fueron:

T1: hora en que el cliente se sienta en la mesa

T2: hora en que el cliente recibe la carta

T3: hora en que el cliente realiza su orden

T4: hora en que el mozo registra la orden en el punto de adición

De ello obtuvimos tres variables a observar:

A: Promedio de tiempo entre que el cliente se sienta y recibe la carta

B: Promedio de tiempo entre que el cliente recibe la carta y realiza su orden al mozo
C: Promedio de tiempo entre que el mozo recibe el pedido del cliente y lo ingresa en el punto de adición

T1: hora en que el cliente se sienta en la mesa

T2: hora en que el cliente recibe la carta

T3: hora en que el cliente realiza su orden

T4: hora en que el mozo registra la orden en SIGESPED

De ello obtuvimos tres variables a observar:

A: Promedio de tiempo entre que el cliente se sienta y recibe la carta

B: Promedio de tiempo entre que el cliente recibe la carta y realiza su orden al mozo

C: Promedio de tiempo entre que el mozo recibe el pedido del cliente y lo ingresa en su tableta con SIGESPED

* Medidos en segundos

La información es la siguiente:

	A	B	C
MESA 1	310	280	470
MESA 2	315	302	488
MESA 3	332	318	505
MESA 4	300	282	471
MESA 5	315	295	487
MESA 6	335	327	518
MESA 7	321	312	500
MESA 8	340	332	520
MESA 9	320	312	501
MESA 10	315	308	502
MESA 11	330	310	505
MESA 12	340	333	518
MESA 13	310	292	485
MESA 14	302	294	485
MESA 15	308	292	485
MESA 16	315	295	493
MESA 17	312	310	508
MESA 18	318	315	508
MESA 19	332	322	510
MESA 20	315	300	495

Luego de analizar los datos por coeficiente de correlación obtuvimos los siguientes resultados:

Corr(A,B)	0.90273342
Corr(B,C)	0.97057141
Corr(A,C)	0.85493546

Esto indica que las variables están fuerte relacionadas en el horario observado.

DESPUÉS DE IMPLEMENTAR

C. Observaciones marzo en horario normal

Como muestra tenemos 20 mesas del local Dasso en horario normal.

Los datos observados fueron:

T1: hora en que el cliente se sienta en la mesa

T2: hora en que el cliente recibe la carta

T3: hora en que el cliente realiza su orden

T4: hora en que el mozo registra la orden en SIGESPED

De ello obtuvimos tres variables a observar:

A: Promedio de tiempo entre que el cliente se sienta y recibe la carta

B: Promedio de tiempo entre que el cliente recibe la carta y realiza su orden al mozo

C: Promedio de tiempo entre que el mozo recibe el pedido del cliente y lo ingresa en su tableta con SIGESPED

* Medidos en segundos

La información es la siguiente:

	A	B	C
MESA 1	108	304	34
MESA 2	113	301	34
MESA 3	119	329	36
MESA 4	120	322	36
MESA 5	115	298	34
MESA 6	106	297	33
MESA 7	103	280	33
MESA 8	97	265	32
MESA 9	125	329	35
MESA 10	113	306	34
MESA 11	103	270	31
MESA 12	108	282	32
MESA 13	122	329	37
MESA 14	107	298	35
MESA 15	125	339	38
MESA 16	110	281	33
MESA 17	98	262	30
MESA 18	119	321	37
MESA 19	104	310	33
MESA 20	109	274	32

Luego de analizar los datos por coeficiente de correlación obtuvimos los siguientes resultados:

$\text{Corr}(A,B)$ 0.88423342
 $\text{Corr}(B,C)$ 0.91921901
 $\text{Corr}(A,C)$ 0.86245948

Esto indica que las variables están fuerte relacionadas en el horario observado.

D. Observaciones marzo en hora punta

Como muestra tenemos 20 mesas del local Dasso en hora punta.

Los datos observados fueron:

T1: hora en que el cliente se sienta en la mesa

T2: hora en que el cliente recibe la carta

T3: hora en que el cliente realiza su orden

T4: hora en que el mozo registra la orden en el punto de adición

De ello obtuvimos tres variables a observar:

A: Promedio de tiempo entre que el cliente se sienta y recibe la carta

B: Promedio de tiempo entre que el cliente recibe la carta y realiza su orden al mozo

C: Promedio de tiempo entre que el mozo recibe el pedido del cliente y lo ingresa en el punto de adición

* Medidos en segundos

La información es la siguiente:

	A	B	C
MESA 1	312	302	40
MESA 2	326	315	41
MESA 3	320	311	42
MESA 4	310	299	40
MESA 5	315	306	40
MESA 6	310	301	39
MESA 7	300	292	39
MESA 8	311	305	39
MESA 9	339	327	42
MESA 10	327	315	41
MESA 11	314	307	41
MESA 12	321	311	41
MESA 13	341	333	42
MESA 14	320	322	41
MESA 15	334	330	41
MESA 16	320	308	40
MESA 17	336	330	42
MESA 18	313	300	40
MESA 19	302	296	39
MESA 20	331	310	41

Luego de analizar los datos por coeficiente de correlación obtuvimos los siguientes resultados:

Corr(A,B) 0.93021104

Corr(B,C) 0.82314237

Corr(A,C) 0.85428427

Esto indica que las variables están fuerte relacionadas en el horario observado.

También podemos comparar los índices de correlación de los datos observados antes y después de la implementación de SIGESPED.

	Febrero Hora normal	Febrero Hora punta	Marzo Hora normal	Marzo Hora punta
Corr(A,B)	0.94316718	0.90273342	0.88423342	0.93021104
Corr(B,C)	0.91991622	0.97057141	0.91921901	0.82314237
Corr(A,C)	0.86206655	0.85493546	0.86245948	0.85428427

COMPARACIÓN ANTES vs DESPUÉS

Los índices de correlación se mantienen fuertes en los cuatro casos evaluados.

Otro punto a tener en consideración es que la variable C disminuyó drásticamente la cantidad de segundos entre los periodos de febrero a marzo tanto en hora normal como en hora punta.

(Tiempo promedio en segundos)	ANTES Hora normal	DESPUES Hora normal	ANTES Hora punta	DESPUÉS Hora punta
Variable A	111	111.2	319.25	320.1
Variable B	299.25	299.85	306.55	311
Variable C	197.65	33.95	497.7	40.55

En horario normal disminuye en 82% y en hora punta disminuye en 91%.

1.10.3. Simulación de costos en apertura de nuevos locales Delicass

Para este punto nos apoyaremos en tablas ya presentadas. La primera de ellas es respecto a los costos actuales de implementación de Delicass (tabla 2):

Descripción	Cantidad	Precio (dólares)	Dcto	Importe
Licencia Inforest Base	1.00	1,000.00	0.00	1,000.00
Licencia Inforest Punto de Venta	1.00	650.00	0.00	650.00
Valor de implementación	1.00	495.00	0.00	495.00
Equipo servidor y caja	1.00	930.00	0.00	930.00
Puntos de adición	1.00	810.00	0.00	810.00

Tabla 2: Precios implementación Inforest

La segunda es la tabla de presupuesto del proyecto de implementación de SIGESPED.

Nombre del Recurso	Cantidad	Precio por Hora	Horas de Trabajo Mensual	Mensual (Cantidad)	Costo Total
Costo de Desarrollo			480 horas	03 meses	
Diseñador de APP	01	S/. 30	45 horas		S/. 1,350
Desarrollador – Programador	01	S/. 30	300 horas		S/.9,000
QA-Tester	01	S/. 35	90 horas		S/. 3,150
Documentador de la APP	01	S/. 20	45 horas		S/. 900
Hardware: Laptop	02	S/. 3.5		3 meses	S/. 1,680
Servicio de Internet Wifi	01	S/. 1.25		3 meses	S/. 540
Total, Costo de Desarrollo					S/. 16,620

La empresa Delicass tiene dos opciones de implementación de puntos de venta para sus locales:

OPCIÓN UNO:

	Costo
Licencias	\$ 2145.00
Equipos	\$ 1740.00

Únicamente el sistema Inforest son \$ 3885.00 dólares como mínimo por local como mínimo (solo un punto de adición).

OPCIÓN DOS:

Sistema Inforest + SIGESPED como punto de adición

	Costo
Licencias Inforest	\$ 1495.00
Implementación SIGESPED	\$ 5036.00
Equipos (10 tablets)	\$ 1118.00

Son \$7649 dólares considerando que el pago de implementación de SIGESPED es por sólo una vez. A partir del segundo local, solo serían \$2613 dólares.

Haciendo un cuadro comparativo:

	Local 1	Local 2	Local 3	Local 4	Local 5	Local 6	Local 7
Opc. 1	\$ 3885	\$ 3885	\$ 3885	\$ 3885	\$ 3885	\$ 3885	\$ 3885
Opc. 2	\$7649	\$ 2613	\$ 2613	\$ 2613	\$ 2613	\$ 2613	\$ 2613
Ahorro	\$ -3764	\$ 1272	\$ 1272	\$ 1272	\$ 1272	\$ 1272	\$ 1272

Como se puede apreciar a partir del 4to local recién la empresa Delicass verá un ahorro, el cual se hará mucho más visible a partir del 5to nuevo local.

CONCLUSIONES

- Se puede diseñar una aplicación Android que escriba, lea, edite y elimine en una base de datos, por medio de web services en lenguaje Php.
- La aplicación puede trabajar en paralelo con el sistema Inforest
- Los coeficientes de correlación en ambos informes indican que existe una fuerte relación entre las variables observadas.
- El promedio de tiempo de atención en la toma de pedidos disminuye en el periodo de implementación de SIGESPED.
- Se demuestra que Delicass tendrá un ahorro de más de \$ 1000 a partir del 5to nuevo local.

REFERENCIAS BIBLIOGRÁFICAS.

- Muñoz, M. (2017). UNIVERSIDAD INCA GARCILAZO DE LA VEGA. Desarrollo de una aplicación móvil para la realización de reservas y tomas de órdenes en el restaurante LongHorn.
- Minchola, R. & Zumarán, B. (2016). UNIVERSIDAD PRIVADA DEL NORTE. Sistema web y móvil para la mejora de recepción de pedidos en el proceso delivery en la empresa Don Belisario.
- Ramos, C. & Suarez, E. (2015). UNIVERSIDAD DE CÓRDOBA. Diseño de una plataforma web y una aplicación móvil para la administración y generación de pedidos en los restaurantes del municipio de Planeta Rica Córdoba
- Europa Press (Enero 2013) Atecrea crea un sistema en la que el cliente de un bar puede hacer su pedido sin esperar por el camarero. Recuperado el 20 de diciembre de 2018 de <https://www.20minutos.es/noticia/1712412/0/>
- Díaz De Orbe, G. (Setiembre de 2010). UNIVERSIDAD PONTIFICIA COMILLAS. Obtenido de <http://www.iit.upcomillas.es/pfc/resumenes/4c80a8886c773.pdf>
- Burga Cruz, J., & Huamán Montero, Y. L. (2013). Scribd. Obtenido de <https://es.scribd.com/doc/138210152/Tesis-Delivery-y-Pago-Movil-Parte-1ver3-Correccion#scribd>
- Morales Guevara, P. D., Chavira , G., & Álvarez , E. (20 de Junio de 2014). Diseño de Un Modelo de Aplicación Móvil En Entorno Android. EAE.

- Ponce, D. (2003) Contribución al desarrollo de un entorno seguro de m-commerce. Publicado el 2 de Setiembre de 2002 y obtenido el 14 de Abril del 2014, desde: <http://www.tdx.cat/handle/10803/7029>.