



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA

“DESARROLLO Y DISTRIBUCIÓN DE UNA APLICACIÓN MÓVIL
PARA AUTOMATIZAR EL COBRO DE PEDIDOS DE
DISTRIBUIDORES DE COMIDA RÁPIDA POR MEDIO DE
RECONOCIMIENTO DE VOZ”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA ADMINISTRATIVA

PRESENTA:

JOSÉ FRANCISCO VERGARA ESPINOZA

ASESOR:

M. EN C.C. JUAN CARLOS MORENO SÁNCHEZ

REVISORES:

DR. EN C. ADRIAN TRUEBA ESPINOSA

M. EN C.C. ANA LUISA MARTÍNEZ ÁVIDA

TEXCOCO, ESTADO DE MÉXICO, JULIO DE 2018.

Texcoco, Méx. a 29 de mayo de 2018.

Asunto: Etapa de digitalización

M. EN C. E. VIRIDIANA BANDA ARZATE
SUBDIRECTORA ADADÉMICA DEL
CENTRO UNIVERSITARIO UAEM
TEXCOCO.

PRESENTE

AT'N M. EN C. LETICIA ARÉVALO CEDILLO.

RESPONSABLE DEL DEPARTAMENTO DE TITULACIÓN

Con base en las revisiones efectuadas al trabajo escrito titulado “DESARROLLO Y DISTRIBUCIÓN DE UNA APLICACIÓN MÓVIL PARA AUTOMATIZAR EL COBRO DE PEDIDOS DE DISTRIBUIDORES DE COMIDA RÁPIDA POR MEDIO DE RECONOCIMIENTO DE VOZ” que para obtener el título de Licenciado en Informática Administrativa presenta el sustentante José Francisco Vergara Espinoza, con número de cuenta 1225191, se concluye que cumple con los requisitos teórico-metodológicos por lo que se le otorga el voto aprobatorio para su sustentación, pudiendo **continuar con la etapa de digitalización** del trabajo escrito.

ATENTAMENTE


M. en C. Ana Luisa Martínez Avido
NOMBRE Y FIRMA DEL REVISOR


Dr. Adrián Irueba Espinosa
NOMBRE Y FIRMA DEL REVISOR


M. en C. Juan Carlos Moreno Sánchez
NOMBRE Y FIRMA DEL DIRECTOR

c.c.p. JOSÉ FRANCISCO VERGARA ESPINOZA.

c.c.p. Director: M EN C. C. JUAN CARLOS MORENO SÁNCHEZ.

c.c.p. Titulación / M. EN C. LETICIA ARÉVALO CEDILLO.

AGRADECIMIENTOS

Agradezco a mis padres por haberme apoyado moral y económicamente para terminar mis estudios universitarios.

Agradezco el apoyo de mi director de tesis, quien me ha formado en mi vida académica, personal y profesional.

Agradezco a la UAEM, dado que me formó académicamente y gracias a sus docentes tuve una nueva perspectiva de la vida.

ÍNDICE

RESUMEN	11
II INTRODUCCIÓN	12
III PLANTEAMIENTO DEL PROBLEMA	14
IV JUSTIFICACIÓN.....	16
V OBJETIVOS	18
5.1 Objetivo general.....	18
5.2 Objetivos específicos.....	18
VI MARCO REFERENCIAL	19
6.1 Bill Calculator.....	19
6.2 Easy Sales.....	19
6.3 Appstaurante.....	20
6.4 Comida a domicilio.....	20
6.5 Restalo.....	21
6.6 Hellofood.....	21
6.7 Alacarta app.....	21
6.8 Contraste de aplicaciones.....	22
VII MARCO TEÓRICO.....	23
7.1 Las PyMES en México.....	23
7.2 Smartphone.....	25
7.3 Sistema de información.....	25
7.3.1 Sistemas transaccionales.....	27
7.3.2 Sistemas de apoyo la toma de decisiones.....	28
7.3.3 Sistemas de información estratégicos.....	29
7.4 Tecnologías de la información.....	31
7.5 Sistema Operativo.....	32
7.5.1 Sistema Operativo “Android”.....	33
7.5.2 Comparación entre Android y iOS.....	34
7.6 Lenguaje de programación.....	35
7.6.1 Objective-C.....	35

7.6.2 Java.....	36
7.7 Bases de datos.	36
7.7.1 Sistemas gestores de bases de datos.	37
7.7.2 SQLite.	38
7.7.3 Modelo relacional.....	41
7.7.4 Modelo entidad relación.....	42
7.7.5 Modelo entidad relación extendida.	42
7.7.6 Normalización de bases de datos.	42
7.8 Inteligencia Artificial (IA).	43
7.9 Reconocimiento automático de voz.....	45
7.9.1 Historia de los sistemas RAH.....	45
7.9.2 Caracterización de los sistemas de reconocimiento de voz.....	46
7.9.3 Fuentes de variabilidad de las señales de voz.	47
7.9.4 Reconocimiento de voz en la nube de Google.	48
7.9.5 Modelos ocultos de Markov para el reconocimiento automático del habla.	50
7.10 Lenguaje Formal	51
7.10.1 Analizador Sintáctico.	51
7.10.2 Analizador Lexicográfico.....	52
7.10.3 Tokenización.	52
7.11 Autómatas.	52
7.11.1 Diagrama de transiciones.	53
7.11.2 Tabla de transiciones.....	53
7.12 Dispositivos móviles.....	54
7.13 Aplicaciones Móviles.	54
7.14 Metodologías de desarrollo ágil.....	54
7.14.1 Metodología Crystal.....	56
7.14.2 Scrum.	57
7.15 Patrones de diseño.	60
7.15.1 MVC.....	60
7.15.2 DAO.	60
7.15.3 DTO.....	61
7.15.4 Capa de acceso a datos (DAL).	61

7.15.5 Lógica de negocio (Business Layer).....	61
VIII METODOLOGÍA.....	62
8.1 Descripción del negocio (caso práctico).....	63
8.2 Problemática del negocio.....	63
8.3 Requisitos generales.	64
8.4 Normalización de base de datos.....	64
8.1 Iteración 1.	66
8.1.1 Planificación de la iteración 1.	67
8.1.2 Ejecución de la iteración 1.	70
8.1.3 Entrega de la primera iteración.	72
8.1.4 Feedback.....	72
8.2 Iteración 2.	73
8.2.1 Planificación de la iteración 2.	73
8.2.2 Ejecución de la iteración 2.	75
8.2.3 Entrega de la iteración 2.....	82
8.2.4 Feedback.....	83
8.3 Iteración 3.	83
8.3.1 Planificación de la iteración 3.	83
8.3.2 Ejecución de la iteración 3.	85
8.3.3 Entrega de la iteración 3.....	93
8.3.4 Feedback.....	93
8.4 Iteración 4.	94
8.4.1 Planificación de la iteración 4.	94
8.4.2 Ejecución de la iteración 4.	95
8.3.3 Entrega de la iteración 3.....	97
8.3.4 Feedback.....	98
8.5 Publicación de la app en Play Store.	98
8.6 Pruebas.....	99
8.6.1 Pruebas de tiempo.	99
8.6.2 Prueba de costos.	99
8.6.3 Satisfacción del usuario.....	100
IX RESULTADOS.....	101

9.1 Cuadro de comparación de tiempos “Humano vs App”.....	101
9.2 Costos.....	102
9.3 Encuesta de satisfacción del usuario.....	103
9.4 Publicación de la app en el repositorio de Play Store.....	106
X DISCUSIÓN.....	107
XI CONCLUSIÓN	109
XII BIBLIOGRAFÍA	111
XIII ANEXOS	115
13.1 Anexo A, encuesta a PYMES encargadas de la preparación y venta de alimentos y bebidas.	115
13.2 Anexo B, encuesta de satisfacción de usuarios.....	116

ÍNDICE DE FIGURAS

Figura 1 Clase que extiende a SQLiteOpenHelper.....	41
Figura 2. Sistemas de reconocimiento de voz basado en redes neuronales.....	50
Figura 3. Ejemplo de transición de estados en un modelo oculto de Márkov.....	51
Figura 4. Diagrama de la metodología Scrum.....	58
Figura 5. Normalización de la base de datos utilizando la 1FN.....	65
Figura 6. Normalización de la base de datos utilizando la 2FN.....	66
Figura 7 App "CalculaTacos" ejecutada (iteración 1).....	71
Figura 8 App "CalculaTacos" con datos "iteración 1".	71
Figura 9 App "CalculaTacos" ejecutada y con datos (iteración 2).....	76
Figura 10 App "CalculaTacos" Activity Productos (iteración 2).....	77
Figura 11 App "CalculaTacos" Activity Productos-Ver Productos (iteración 2).....	78
Figura 12 App "CalculaTacos" Activity AgregarProducto (iteración2).....	78
Figura 13 App "CalculaTacos" Activity EliminarProducto (iteración2).....	79
Figura 14. Algoritmo utilizado para reconocer las instrucciones de voz que el usuario envía. Fuente propia.....	80
Figura 15 App "CalculaTacos" ejecutada y con datos (iteración 3).....	86
Figura 16 App "CalculaTacos" Activity Productos (iteración 3).....	87
Figura 17 App "CalculaTacos" Activity Productos-Ver Productos (iteración 3).....	88

Figura 18 App "CalculaTacos" Activity AgregarProducto (iteración 3).	89
Figura 19 App "CalculaTacos" Activity EliminarProducto (iteración 3).....	90
Figura 20 App "CalculaTacos" Actualizar Producto (iteración 3).	91
Figura 21 Logotipo de app "CalculaTacos"	92
Figura 22. Activity Principal con elemento Spinner.	95
Figura 23. Activity Reporte, Iteración 4.....	96
Figura 24. Activity Pago, Iteración 4.	97
Figura 25 Tiempo de respuesta por cada método utilizado en la comparación de tiempos.	102
Figura 26 Aceptación de la app en cuanto a funcionalidad.....	104
Figura 27 Aceptación de la app en cuanto a su usabilidad.....	104
Figura 28 Nivel de efectividad de la app.....	105

ÍNDICE DE TABLAS

Tabla 1. Comparación entre Android y iOS.....	34
Tabla 2. Tabla inicial requerimientos de la base de datos.....	65
Tabla 3. Roles ocupados dentro de la metodología Scrum.....	67
Tabla 4. Historias de usuario (Iteración 1)	69
Tabla 5. Historias de usuario (Iteración 2).	74
Tabla 6. Sintaxis para utilizar la app.	81
Tabla 7. Tabla ficticia de precios.....	81
Tabla 8. Array "palabras".	82
Tabla 9. Historias de usuario (Iteración 3).	84
Tabla 10. Historias de usuario (Iteración 4).	94
Tabla 11. Humano vs App (tiempo).	101
Tabla 12. Errores cometidos en cada uno de los métodos.....	102

RESUMEN

El presente proyecto de investigación brinda una solución viable a una de las problemáticas que tienen las PYMES encargadas de la preparación de alimentos y bebidas, como lo es el control de las ventas. Con base en los resultados de una encuesta (Ver anexo A) realizada a este tipo de empresas, se detectó que en su mayoría tienen problemas en cuanto a la efectividad del cálculo en cada una de las ventas realizadas durante las horas pico.

Debido a que el mercado de la telefonía celular está dominado por los dispositivos móviles con sistema operativo android, dicha aplicación está desarrollada para este tipo de dispositivos. Por otra parte, cabe mencionar que la app hace uso del gestor de base de datos llamado SQLite, debido a que fácilmente se adapta a las características de los smartphones y tabletas. Se utilizó una metodología de desarrollo ágil como lo es Scrum, se eligió esta metodología debido a que cuenta con características como su simplicidad, adaptabilidad ante nuevos requerimientos, y que es incremental.

Con un fin práctico se desarrolla la aplicación aplicada a un caso práctico “Taquería Espinoza” como resultado se generó la aplicación móvil llamada CalculaTacos., ésta tiene el objetivo de llevar el control de ventas de los distribuidores de comida rápida, haciendo uso del reconocimiento de voz. Además de automatizar este proceso, la app cuenta con un módulo que permite generar reportes históricos para los administradores de estas empresas, lo cual facilitará la toma de decisiones.

Como ya se ha mencionado, la app está diseñada para ser utilizada en cualquier PYME encargada de la preparación de alimentos y bebidas; pero para efectos de medición en este trabajo de investigación, se optó por implementar la app en una taquería con el fin de medir sus resultados en cuanto a tiempo, costo y satisfacción del usuario.

II INTRODUCCIÓN

A medida que transcurre el tiempo y la tecnología avanza, para cualquier empresa es indispensable estar a la vanguardia en cuanto al uso de la tecnología en todos los ámbitos de negocio. Dado que la competencia cada día es más difícil, es imprescindible para aquellas empresas que aspiren a prosperar, estar un paso delante de sus competidores. Una de las formas de poder destacar es optimizando sus recursos, empleando técnicas innovadoras de realizar sus procesos, en otras palabras, se necesita innovación. En este sentido, la competitividad de una nación depende de la capacidad de su industria para innovar y mejorar. Esto obliga a las empresas a innovar, de caso contrario serán alcanzadas por sus competidores (Porter, 1990).

La metodología Scrum ágil posee características como: adaptabilidad ante requisitos cambiantes, productos funcionales en cada iteración, armonía entre el cliente y el equipo de desarrolladores, entre otras. Se adoptó este tipo de metodología ya que proporciona varias ventajas en cuanto la optimización de tiempo y contribuye a que el desarrollo de software se lleve de la mano con el cliente, lo cual garantiza que el producto final cumpla con las expectativas del cliente.

El trabajo se encuentra dividido en doce capítulos, en los cuales se redacta detalladamente desde los motivos que hicieron este trabajo posible, pasando por la planeación tomando en cuenta los objetivos a los que se pretendió llegar, posteriormente se retomaron algunos conceptos importantes dentro del marco teórico, mientras que en el marco referencial se tomó en cuenta los trabajos que comparen alguna similitud o filosofía a éste. El octavo capítulo se enfoca específicamente a explicar el desarrollo de la aplicación móvil, haciendo referencia a cada una de las tareas que estipula Scrum. En la parte final de la tesis se muestran los resultados obtenidos al llevar a cabo el desarrollo móvil, para finalmente realizar una discusión y establecer las conclusiones correspondientes a las que se llegó. La presente tesis brinda una solución a un problema tan común dentro de las PyMES mexicanas dedicadas a la preparación de alimentos y bebidas, dicho problema radica principalmente en una de las tareas que forman parte del proceso de venta, como lo es el costo total generado por la venta de productos a un cliente en específico. En la mayoría de este tipo de empresas, la tarea de

calcular la cuenta tiende a generar problemas debido a las técnicas utilizadas, que por lo general producen errores aritméticos o consumen demasiado tiempo. Además de que dicha app tiene la capacidad de generar reportes históricos a sus usuarios para una mejor toma de decisiones.

Para el caso práctico, se creó un sistema que permite dar soporte en el área de ventas. además de que la metodología nos permite llevar el desarrollo de estas aplicaciones a solucionar cualquier otro caso práctico con un mínimo de ajuste para presentar un modelo funcional y de esta forma no iniciar desde cero con el fin de poder dar soluciones particulares a diversas empresas dentro de este sector.

III PLANTEAMIENTO DEL PROBLEMA

Es imperativo el uso de las tecnologías de la información y comunicación en las PyMES, ante los nuevos retos que les presenta el siglo XXI, siendo la innovación un elemento clave que explica la competitividad para las empresas como el medio para incrementar su mercado y por ende beneficios (COTEC, 1999).

Se concibe que en México las PyMES no están haciendo uso de las TICs debido a diversos factores; entre los más destacados se encuentra la falta de cultura tecnológica e inversión que se destina por el alto costo y la reducida oferta que hay en el mercado (Ravenna & González, 2004). Además de que en ocasiones estas empresas no cuentan con personal suficiente para realizar todas sus actividades de forma equitativa. Ya que por lo regular una persona toma más de dos roles laborales.

Los avances tecnológicos y económicos, han originado la existencia de una gran diversidad de empresas, pero básicamente se pueden clasificar de acuerdo a su actividad o giro: industriales, comerciales y de servicio (Bautista, s.f.). En cuanto a los negocios encargados de la preparación de alimentos y bebidas, se localiza en las empresas comerciales, debido a que son aquellas las intermediarias entre productor y consumidor.

Las PyMES encargadas de la preparación y venta de alimentos tales como: taquerías, restaurantes, fondas, entre otras son de las más comunes en México. Y es que este tipo de negocios son de los más socorridos ya que satisfacen una de las necesidades primarias del ser humano como lo es la alimentación.

En el municipio de Texcoco de Mora y alrededores, existe una gran necesidad en cuanto a la automatización de los procesos de venta. Con el paso del tiempo y aumento de demanda de alimentos preparados, los encargados de estos negocios se han enfrentado a problemas en cuanto a la exactitud al realizar la cuenta total de consumo a un cliente, además del tiempo que demoran en hacer cada una de las cuentas de manera mental o haciendo uso de calculadoras estándar.

Si bien es cierto que las personas encargadas de atender a los clientes tienen una gran habilidad para hacer cuentas mentalmente, es normal que con la presión de atender a

diferentes clientes y el estrés generado por realizar diferentes tareas al mismo tiempo lleguen a cometer errores en el cobro de cuentas. Este error repercute en el corto plazo en la pérdida de clientes; mientras que en el largo plazo se refleja en falta de rentabilidad para la empresa, lo cual genera desempleo y falta de productividad para nuestro país.

IV JUSTIFICACIÓN

El presente proyecto tiene la finalidad de desarrollar una aplicación móvil para dispositivos android utilizando el reconocimiento automático de voz para automatizar el cobro de pedidos de distribuidores de comida rápida. Debido a que hoy en día, en México existen aproximadamente 4 millones 15 mil unidades empresariales de las cuales 99.8% son PyMES que generan 52% del Producto Interno Bruto (INEGI, Censos económicos 2014, La industria restaurantera en México., 2016). Se debe de considerar que el uso de la tecnología por parte de las PyMES es imprescindible para brindarles una ventaja competitiva en el mercado. En este sentido, los distribuidores de comida rápida forman parte de las PyMES en México; sin embargo, la mayoría de estas PyMES no hacen uso de la tecnología con el fin de apoyarlas en sus procesos. Por tal motivo una de las necesidades prioritarias para las PyMES en la segunda década del siglo XXI, es la adopción de nuevas tecnologías para poder competir en el mercado en el que se encuentran.

Uno de los Motivos que han hecho que las PyMES mexicana hagan poco uso de la tecnología, es la falta de inversión, puesto que, de acuerdo a los resultados de la encuesta realizada, el 95% de los establecimientos encuestados estarían dispuestos a implementar algún tipo de tecnología en sus procesos, especialmente utilizando un dispositivo electrónico tan común como lo es el smartphone de uso diario. Considerando que al hacer uso de su smartphone evitarían costos como el hardware en una computadora de escritorio. Por otra parte, las personas que laboran en estos negocios están más abiertos a utilizar su smartphone como herramienta de trabajo porque les parece muy intuitivo su uso, tiene un tamaño manejable e interactúan con él todos los días, por lo que aprender a usar una app nueva no les representa mayor problema.

Con el paso del tiempo y las circunstancias socioeconómicas por las que atraviesa México, han hecho que comer fuera de casa sea una necesidad para los mexicanos. Es por ello que este tipo de negocios son visitados diariamente por gran parte de la población mexicana. Esto origina que en determinados horarios este tipo de negocios se enfrente al reto de atender a un gran número de clientes a la vez, presentando problemas al momento de cobrar la cuenta a sus clientes, lo que dificulta la realización de operaciones mentales. Siendo cierto que los

encargados tomar pedidos y cobrar cuentas poseen una buena habilidad para este tipo de operaciones, pero en ocasiones no es lo suficientemente efectiva al enfrentarse al estrés derivado de la presión por atender otros pedidos. Esto a su vez repercute en la disminución de ventas y dinero. En este sentido, la aplicación que se pretende desarrollar ayudará a calcular las cuentas de los productos consumidos por cada cliente con el 0.0% de margen de error en el resultado al hacer un uso correcto de la app.

Por otra parte, las personas encargadas del cobro de los pedidos suelen utilizar una calculadora genérica o en su defecto la calculadora de su smartphone, sin embargo, esta práctica no parece ser la más óptima puesto que lleva tiempo calcular individualmente el subtotal de cada producto y posteriormente calcular el total de los productos vendidos. Es por ello que el producto del presente proyecto será capaz de realizar las operaciones aritméticas necesarias en una fracción del tiempo del que requieren los métodos tradicionales.

En otras palabras, el presente proyecto tiene la finalidad de desarrollar una aplicación móvil, la cual funcionará como calculadora con reconocimiento de voz, para calcular las cuentas totales generadas por los pedidos en los negocios encargados de la venta de comida rápida; la cual ayudará a hacer más eficiente el proceso de cobro total de productos, sin mencionar la reducción de tiempo al realizar las cuentas. Así mismo la app tendrá la capacidad de generar un reporte del historial de cada venta realizada en la aplicación. Para que posteriormente, el administrador de la PyME que haga uso de la app pueda tomar decisiones con base en los reportes.

Además de optimizar el tiempo de cobro hacia los clientes y ser una herramienta para la toma de decisiones, la aplicación móvil tendrá el objetivo de brindar una ventaja competitiva al negocio que haga uso de ella; dado que dicha aplicación tendrá la capacidad de ejecutar operaciones aritméticas relacionadas con los productos y precios de cada una de las cuentas mediante comandos de voz. Haciendo que el encargado de cobrar los pedidos ejecute esta operación en segundos; lo cual es muy importante en momentos en donde los clientes necesiten una atención más rápida. Además de que haciendo un uso correcto la aplicación móvil permitirá que el margen de error sea nulo a la hora de hacer el cobro de los pedidos.

V OBJETIVOS

5.1 Objetivo general.

Desarrollar un sistema para el control de las ventas empleando reconocimiento de voz, aplicado a los distribuidores de comida rápida.

5.2 Objetivos específicos.

- Planeación del proyecto.
- Desarrollo de la base de datos.
- Desarrollo del sistema en Android.
- Implementar el sistema en un caso práctico Taquería Espinoza.
- Planeación de las pruebas y liberación de la aplicación.
- Publicación de la aplicación, en un medio electrónico para su distribución.

VI MARCO REFERENCIAL

Actualmente existen diferentes sistemas de toma de pedidos y de control de ventas para las diferentes empresas encargadas de la venta comida. Estos van desde sistemas que reservan mesas y pedidos de manera online, hasta simples dispositivos táctiles con el menú disponible; instalados en los establecimientos. Sin lugar a dudas estos sistemas han beneficiado de manera significativa a los negocios que se han dado a la tarea de implementarlos. A continuación, se describirán algunos de los sistemas utilizados hoy en día por restaurantes y cadenas de comida rápida.

6.1 Bill Calculator.

Es una aplicación móvil exclusiva de dispositivos android. Los usuarios potenciales de esta aplicación móvil son aquellas personas que acuden a establecimientos de comida y quieren saber el total a pagar por los productos recibidos. El usuario tiene la capacidad de generar una cuenta y agregar cada producto con su respectivo precio y cantidad. Cabe mencionar que es necesario ingresar el nombre del producto, la cantidad y el costo cada vez que se quiere calcular el precio de un producto. Por lo que es tedioso estar ingresando esos tres datos cada vez que se acude a un negocio de venta de comida (Cheung, 2018).

6.2 Easy Sales.

EasyNet es una empresa colombiana que intenta solventar las necesidades de comunicación mediante dispositivos móviles para el sector corporativo. Esta empresa a su vez posee un tipo de tecnología llamado Easy Sales. Esta es una solución móvil para la automatización de fuerza de ventas, basada en las mejores prácticas del mercado. Es una herramienta de trabajo eficiente para la consulta, almacenamiento, análisis y visualización de toda la información necesaria para el ejecutivo de ventas en campo y el personal encargado de su supervisión.

Easy Sales no es necesariamente un tipo de tecnología que se encarga de brindar soluciones de comunicación móvil específicamente a empresas encargadas de la venta de alimentos; sin

embargo esta es una de las funciones que ha implementado en diferentes restaurantes, en donde se ocupan las diferentes capacidades gráficas, visuales, de almacenamiento y comunicación que puede tener un dispositivo móvil con la posibilidad de tener toda la información de los sistemas de ventas y pedidos al instante y en cualquier lugar (Narvin, 2018).

6.3 Appstaurante.

Esta aplicación es diseñada según la necesidad específica del cliente, de esta manera se realiza una solución por cada restaurante, es decir es una app a la medida, The Food Marketing es una agencia de marketing, publicidad y diseño que se especializan en restaurantes, la cual trata de asesorar y prestan servicios a propietarios de restaurantes, pizzerías, hoteles, y otras formas de establecimientos gastronómicos (Appstaurante, 2018) .

Appstaurante consiste en brindar al propietario del establecimiento una herramienta en la permite a que los clientes puedan hacer su uso y les facilite interactuar y realizar las siguientes funciones:

- Visualizar platos disponibles.
- Validar el contenido de cada plato.
- Ver precio de productos.
- Ver imágenes de los diferentes platos.
- Realizar reservas.
- Realizar pedidos
- Ver estado del pedido
- Historial de compra.

6.4 Comida a domicilio.

Esta App fue desarrollada para las provincias españolas, permite realizar una búsqueda e informar un número de contacto, de este modo puede remitirse para realizar el pedido (indalquiler, 2018); esta aplicación está realizada por Gonzalo de Córdoba y tiene las siguientes funcionalidades:

- Búsqueda por ciudad.
- Elegir tipo de comida.
- Encontrar restaurante.
- Brindar información de restaurante.

6.5 Restalo.

Esta herramienta también está diseñada para España, permite realizar búsquedas de platos de una gran lista de restaurantes (TheFork, 2018), y adicional a esto, hacer reservas; las principales funcionalidades de esta herramienta son:

- Encontrar restaurantes.
- Descubrir nuevos restaurantes.
- Opinar sobre cualquier restaurante.
- Realizar reservas.
- Acumular puntos y canjearlos por cenas.

6.6 Hellofood.

Esta aplicación está disponible para países como Argentina, Brasil, Chile, México, Perú y Colombia, teniendo en cuenta, que para este, está disponible en las ciudades de Bogotá, Cali y Medellín; esta app funciona tomando como referencia la posición que se tiene en el momento, por medio del GPS, dando a conocer los restaurantes más cercanos, y así poder realizar cualquier tipo de pedido (24, 2018).

6.7 Alacarta app.

Es una aplicación móvil especializada en mostrar la oferta gastronómica de restaurantes y bares; actualmente se ha enfocado en lugares de la ciudad de Pereira, Colombia; abarcando gran cantidad de espacios que brindan diversión, esparcimiento y por supuesto comidas y bebidas.

De esta manera, Alacarta permite visualizar en un mapa real los sitios exactos de la ciudad en donde están ubicados los puntos de venta y así sus usuarios estarán satisfechos al tener al alcance de su mano toda la información, podrán encontrar los lugares con mayor facilidad y conocer los productos y servicios (Alacarta, 2018).

6.8 Contraste de aplicaciones.

Como se ha descrito anteriormente, existe una gran variedad de aplicaciones móviles dedicadas a la automatización de toma de pedidos, control de ventas e incluso participan en actividades de mercadotecnia. Sin embargo, la mayoría de estas aplicaciones intentan tomar el rol de un punto de venta, lo cual tiene poco que ver con el objetivo de este proyecto. Dado que en ninguno de los ejemplos anteriormente descritos utiliza el reconocimiento automático de voz para realizar alguna de sus funciones. Además de que tienen diferentes funciones complejas, lo cual hace que su uso no sea rápido.

VII MARCO TEÓRICO

7.1 Las PyMES en México.

Actualmente en México las micro, pequeñas y medianas empresas, también llamadas PyMES son el pilar fundamental de la economía nacional debido a su alto impacto en la generación de empleos y la producción nacional. De acuerdo con datos del Instituto Nacional de Estadística y Geografía, en México existen aproximadamente 4 millones 15 mil unidades empresariales de las cuales 99.8% son PyMES que generan 52% del Producto Interno Bruto (Almazán Morales, Méndez Salvatorio, & Lee kim, 2013). Estas estadísticas muestran claramente que las PyMES son el medio que se necesita para impulsar el desarrollo económico y la distribución de la riqueza en nuestro país.

Hoy en día las PyMES se encuentran en giros productivos, de servicios y comercialices; además de que estas compiten en el mercado contra empresas grandes, las cuales por tener más recursos les generan una ventaja competitiva en cuanto a las PyMES afectando a su desarrollo (Capó Vicedo, Exposito, Lanja, & Mesia Buades, 2007).

Entre las problemáticas de las PyMES podemos resaltar que sus inversiones para tecnología, infraestructura, seguridad, entre otras, son de un 0.5% y 0.7%, respectivamente, estancando su propio crecimiento. Por lo anterior resulta lógico que aproximadamente 70% de las Pymes no cuenten con base tecnológica instalada teniendo temor de no poder solventar los costos de inversión y operación que esto implica; aunque también existen razones menos aceptables, como la resistencia al cambio, el desconocimiento de los sistemas de información, la falta de infraestructura y la carencia de visión sobre los beneficios que la tecnología puede traer a una empresa. (Moreno Aroca, 2014).

En este sentido, las PyMES se ven en la necesidad de adoptar la tecnología en sus procesos con la finalidad de tener herramientas suficientes para poder competir en el mercado con empresas grandes o en determinado caso, con PyMES que ya utilicen las tecnologías a su favor.

Existen diferentes formas en las que las PyMES pueden beneficiarse de la tecnología para automatizar sus procesos, tales como: control de inventarios, control de ventas, difusión a

través de redes sociales, entre otras. Sin embargo, en la mayoría de las PyMES de México aún no existe la cultura de adquirir tecnología y en otras ocasiones no cuentan con el presupuesto necesario para montarse en la era digital; lo cual ha repercutido gravemente en la deserción de muchas empresas de este tipo a lo largo de los últimos años. La productividad de las PyMES está sujeta al acervo del capital y conocimiento, al nivel tecnológico con el que opera y a la habilidad directiva para gestionar (Cleri, 2014). Indudablemente estos factores son la clave para el crecimiento de cualquier empresa. Sin embargo, no todas las empresas los toman en cuenta.

Se entiende que las tecnologías de la información son una herramienta que se suma a las tareas de las áreas de una empresa como la contabilidad, las finanzas, la logística, los recursos humanos y las operaciones rutinarias. Los sistemas de información basados en aplicaciones móviles en específico, han cambiado la forma en que operan las organizaciones actuales. A partir de su uso, se logran importantes mejoras en el rendimiento de la empresa, debido a que además de automatizar los procesos, las aplicaciones brindan movilidad a los usuarios.

Desde la aparición del internet un gran número de personas alrededor del mundo empezaron a adquirir dispositivos electrónicos que se conectaran a la red global para disfrutar de todos sus beneficios que abarcaba el intercambio de información. Pero no fue hasta esta última década donde se popularizaron los dispositivos móviles tales como teléfonos inteligentes y tabletas. Este auge tan reciente se debió principalmente a la practicidad del uso de dichos dispositivos debido a que por su tamaño son ideales para llevarlos en el bolsillo del pantalón.

Asimismo, los sistemas operativos que han facilitado el uso de las propiedades de los dispositivos móviles también han evolucionado a la par de dichos dispositivos. Entre los sistemas operativos para dispositivos móviles se encuentran: Android, iOS y Windows Phone. A pesar de que existen millones de dispositivos móviles en todo el mundo, el sistema operativo Android es el que abarca la mayor parte del mercado.

7.2 Smartphone.

Hoy en día la sociedad mexicana vive un fenómeno de exposición y uso creciente de la tecnología celular, en la mayoría de ámbitos. Esto se debe en gran medida a la disminución de costos en los equipos de telefonía celular y al incremento de sus capacidades tecnológicas.

El smartphone o también llamado “teléfono” inteligente, es un dispositivo que cabe en la palma de la mano y cuya aceptación se debe a su gran capacidad de comunicación, procesamiento y almacenamiento, al manejo de contenidos multimedia y a la facilidad de integrarse a las redes inalámbricas (Cruz & López Mortero, 2007).

En México el 57% de personas hace uso de un teléfono inteligente (INEGI, 2016), por lo que es fácil estar en contacto con el reconocimiento de voz, debido a que prácticamente la totalidad de los teléfonos inteligentes incorporan esta tecnología. Por una parte, la aceptación que han tenido los teléfonos inteligentes se debe a que estos cuentan con una pantalla táctil, lo cual facilita la navegación en internet y en general el uso de las diferentes aplicaciones disponibles. Pero una pantalla táctil no lo es todo, en ocasiones los usuarios tienen dificultades al escribir texto, puesto que la pantalla no es lo suficientemente grande como para que las teclas tengan un tamaño considerable. Es aquí donde el reconocimiento de voz surge como una alternativa para redactar cualquier texto de una manera más rápida y eficiente.

7.3 Sistema de información.

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. En un sentido amplio, un sistema de información no necesariamente incluye equipo electrónico (hardware). Sin embargo, en la práctica se utiliza como sinónimo de sistema de información computarizado (Kohen karen & Asín Lares, 2009).

Un sistema de información se compone de los siguientes elementos:

- El equipo computacional: Es el hardware necesario para que el sistema de información opere. Lo constituyen las computadoras y el equipo periférico.

- El recurso humano: Este interactúa con el sistema de información; son las personas que utilizan el sistema, lo alimentan con datos o utilizan los resultados que genera.
- Los datos o información fuente: Son todas las entradas que el sistema necesita para generar la información que se desea.
- El software: Son los programas que procesan los datos de entrada y generan los resultados que se esperan.
- Las telecomunicaciones: Básicamente el hardware y el software que transmiten en forma electrónica texto, datos, imágenes y voz.
- Procedimientos: Incluyen las políticas y reglas de operación, tanto en la parte funcional del proceso de negocio, como los mecanismos para hacer trabajar una aplicación en la computadora.

Cuando los elementos anteriores interactúan entre sí, son capaces de producir información de utilidad para una determinada organización. Para ello, se deben de cumplir cuatro actividades básicas:

1. Entrada de información: Es el ingreso de los datos en el sistema de información. Las entradas pueden ser manuales, las proporciona el usuario, o automáticas, datos o información que provienen o se toman de otros sistemas o módulos. Los típicos dispositivos de entrada de datos a los sistemas de información son las cintas magnéticas, las unidades de disquete, los lectores de código de barras, las plumas ópticas, el escáner, los monitores sensibles al tacto, los lectores de CD-ROM, los lectores de DVD, los USB, el mouse y el teclado.
2. Almacenamiento de información: Es una de las capacidades más importantes de una computadora, puesto que permite al sistema recordar la información guardada en la sesión anterior. Esta información se almacena en estructuras de información denominadas archivos, en su versión simple, y bases de datos, en su modalidad compleja. Los dispositivos de almacenamiento son los discos magnéticos o duros, los discos flexibles o disquetes, los discos compactos (CD-ROM), los discos de alta capacidad (zip), los discos de video (DVD) y los USB.
3. Procesamiento de información: Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos

cálculos trabajan con datos de recién ingreso o con datos ya almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información útil.

4. Salida de información: Es la capacidad de un sistema de información para convertir la información procesada o los datos de entrada en información para el exterior. Las típicas unidades de salida son las impresoras, disquetes, cintas magnéticas, la voz, los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o módulo.

7.3.1 Sistemas transaccionales.

Los sistemas transaccionales tienen como principales características:

- Permitir ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de sistemas de información que se implanta en las organizaciones. Primero sirven para apoyar las tareas a nivel operativo de la organización y, a medida que evolucionan, continúan con los mandos intermedios para posteriormente apoyar a la alta administración.
- Tienen una intensa entrada y salida de información, aunque sus cálculos y procesos suelen ser poco complejos. Estos sistemas requieren de un arduo manejo de datos para poder realizar sus operaciones y, como resultado, generan grandes volúmenes de información.
- Tienen la propiedad de ser recolectores de información, es decir, a través de ellos se cargan las grandes bases de datos para su posterior utilización. Estos sistemas integran cantidades grandes de la información que se maneja en la organización, la cual será empleada después para apoyar a los mandos intermedios y altos.
- Son fáciles de justificar ante la dirección general, dado que sus beneficios son visibles y cuantificables en el corto plazo. El proceso de justificación se realiza con el cotejo de ingresos y costos.

- Son fácilmente adaptables a paquetes de aplicación que se encuentran en el mercado, porque automatizan los procesos básicos que, por lo general, son similares o iguales en otras organizaciones.
- Ejemplos de este tipo de sistemas son facturación, nóminas, cuentas por cobrar, cuentas por pagar, contabilidad general, conciliaciones bancarias, inventarios, abastecimientos, control de distribución, producción, etcétera (Kohen karen & Asín Lares, 2009).

7.3.2 Sistemas de apoyo la toma de decisiones.

El éxito de cada organización depende en gran medida de la calidad de las decisiones que toman sus empleados. Cuando la toma de decisiones implica grandes cantidades de información y mucho procesamiento, las tecnologías de la información hacen más eficiente y eficaz este proceso. Un sistema de soporte de decisiones (DSS), es un sistema de información basado en una computadora diseñado para ayudar a los administradores a elegir una de muchas soluciones alternas a un problema (Oz, 2008). Este tipo de sistemas ofrecen análisis sofisticados y rápidos de inmensas cantidades de datos e información.

Los DSS se deben implementar después de los sistemas transaccionales puesto que estos últimos constituyen su plataforma de información. Por otro lado, la información que generan los DSS suelen ser intensivos en cálculos y escasos en entradas y salidas de información de información, la cual toman de los sistemas transaccionales. Por ejemplo, un modelo de planeación financiera requiere poca información de entrada, pero realiza muchos cálculos durante su proceso.

Los sistemas de apoyo a las decisiones se clasifican en:

- DSS (decision support systems), sistemas de apoyo a la toma de decisiones.
- GDSS (group decision support systems), sistemas para la toma de decisiones en grupo.
- EIS (executive information systems), sistemas de información para ejecutivos.
- EDSS (expert decision support systems), sistemas expertos de apoyo a la toma de decisiones (Kohen karen & Asín Lares, 2009).

Cualquiera de la clasificación de DSS tienen la finalidad de brindar a los individuos encargados de la toma de decisiones es una organización de brindar las bases necesarias con información oportuna y verídica para el logro de sus objetivos planteados.

7.3.3 Sistemas de información estratégicos.

Su función es lograr ventajas que los competidores no poseen, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este sentido, los sistemas estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el comercio electrónico es un sistema estratégico, debido a que da a las empresas elementos para diferenciar sus productos/servicios y así tener ventajas sobre otras compañías que ofrecen productos similares y no cuentan con este servicio. Si una empresa nueva decide entrar al mercado, tendrá que ofrecer este servicio para poder competir.

Los sistemas de información estratégicos tienen las siguientes características:

- Su función no es automatizar procesos operativos ni proporcionar información para apoyar la toma de decisiones. Sin embargo, este tipo de sistemas puede llevar a cabo esas funciones.
- Su desarrollo es dentro de la organización, por tanto, no se adaptan fácilmente a paquetes disponibles en el mercado.
- Su desarrollo típico es en incrementos. Su evolución dentro de la organización inicia con un proceso o función particular y a partir de ahí se agregan nuevas funciones o procesos.
- Apoyan el proceso de innovación de productos y procesos dentro de la empresa, debido a que buscan ventajas respecto de los competidores (Kohen karen & Asín Lares, 2009).

Es importante que se tome en cuenta que las ventajas que se obtienen con estos sistemas no son eternas, es decir, existe un periodo de vigencia que equivale al tiempo que tardan los competidores en alcanzar las diferencias o ventajas obtenidas por el sistema de información estratégico. Cuando esto sucede, los beneficios generados por el sistema de información estratégico se convierten en estándares de la industria, como en el caso del comercio

electrónico, que serán requisitos de infraestructura tecnológica que una compañía debe poseer.

Los negocios, las personas y las organizaciones buscan utilizar la información de manera específica para tomar decisiones sólidas y para resolver problemas. Estas dos últimas prácticas muy relacionadas, son la base de toda empresa exitosa.

Un sistema de información se puede entender como un conjunto de elementos que colaboran entre sí para procesar datos y producir información. En las empresas la mayoría de los sistemas de información con los que cuentan, están integrados por muchos subsistemas con metas secundarias, todas las cuales contribuyen a la meta principal de la organización. Es necesario tomar en cuenta que los elementos por los cuales están formados los sistemas de información pueden ser de diferentes tipos, tales como:

- Datos.
- Hardware.
- Software.
- Telecomunicaciones.
- Personas.
- Procedimientos.

Existen diferentes tipos de sistemas de información, de los cuales una empresa puede hacer uso de acuerdo a sus diferentes jerarquías.

- Sistemas transaccionales.
- Sistemas de apoyo a la toma de decisiones.
- Sistemas estratégicos.

Los sistemas transaccionales, tienen el propósito de realizar procesos que se llevan a cabo en la parte operativa de una empresa. Estos procesos tienen la característica de realizarse de manera rutinaria, además de que no son complejos.

Los sistemas de apoyo a la toma de decisiones se satisfacen de la información resumida que producen los sistemas transaccionales. Con la finalidad de poder brindar un panorama general a los directivos de la empresa, para que de esta forma se puedan tomar las decisiones más acertadas con base en hechos firmes y no en suposiciones o empíricamente.

Los sistemas estratégicos tienen la misión de dotar de alguna ventaja competitiva a una empresa, que sus competidores en el mercado no posean. Las diferentes ventajas pueden ser en costos y servicios diferenciados con clientes y proveedores para incrementar el volumen de ventas.

El uso de los sistemas de información por parte de las empresas dedicadas a la venta de alimentos tales como restaurantes y cadenas de comida rápida; han brindado una ventaja competitiva al ser utilizados en los procesos transaccionales. Cuando se habla de un sistema información basado en aplicaciones móviles que automatice los procesos de toma y cobro de pedidos en establecimientos dedicados a la venta de comida rápida, es imprescindible tener en cuenta los elementos que posee dicho sistema. De acuerdo al problema planteado figuran los siguientes elementos:

- Dispositivo móvil con sistema operativo Android.
- Aplicación móvil.
- Datos de entrada.

7.4 Tecnologías de la información.

Las tecnologías de la información se pueden ver como la base que soporta a los sistemas de formación, debido a que en ellas recaen sus elementos.

Las tecnologías de la información son todas aquellas tecnologías que permiten y dan soporte a la construcción y operación de los sistemas de información, y son tecnologías de hardware, software, de almacenamiento y de comunicaciones. Estas tecnologías forman la infraestructura tecnológica de la empresa, que provee una plataforma en la cual la compañía construye y opera los sistemas de información (Kohen karen & Asín Lares, 2009).

En la actualidad los sistemas de información deben de cumplir con tres objetivos primordiales para las organizaciones:

- Automatizar los procesos operativos.
- Proporcionar información que sirve de apoyo en el proceso de toma de decisiones.
- Logran ventajas competitivas a través de su implantación y uso.

Cuando los sistemas de información automatizan los procesos operativos dentro de una organización se les llama sistemas transaccionales, porque su función primordial consiste en procesar transacciones tales como: pagos, cobros, pólizas, entradas, salidas, etc. El complemento de los sistemas transaccionales son los sistemas enfocados a mejora de la toma de decisiones, cuya función es la manipulación de la información con el fin de apoyar y fundamentar la toma de decisiones. También existen los sistemas de automatización de oficinas, útiles en la mayoría de las empresas para trabajar con documentos. Algunos programas considerados en este tipo de sistemas son: hojas de cálculo, procesadores de texto, herramientas para hacer presentaciones, bases de datos y herramientas para el manejo de imágenes. Otro tipo de sistemas de acuerdo con los objetivos que cumple son los sistemas estratégicos; estos buscan desarrollar en las organizaciones ventajas competitivas a través del uso de las tecnologías de la información.

7.5 Sistema Operativo.

Un sistema operativo es simplemente el intermediario entre la parte física de una computadora (hardware) y el usuario (Tanenbaum, 2009). Por otra parte, un sistema operativo se puede definir como la plataforma de software de un dispositivo, el cual soportará y administrará las aplicaciones para el usuario final.

Las computadoras modernas constan de procesadores, memorias, temporizadores, discos, ratones, interfaces de red, impresoras y una amplia variedad de otros dispositivos. En la perspectiva alterna, el trabajo del sistema operativo es proporcionar una asignación ordenada y controlada de los procesadores, memorias y dispositivos de E/S, entre los diversos programas que compiten por estos recursos (Tanenbaum, 2009).

En este sentido, es necesario mencionar que los dispositivos móviles actuales, tales como tabletas y smartphones pueden ser vistos como computadoras, por lo que estos dispositivos inteligentes cuentan con un sistema operativo que administra sus recursos. A inicios de la segunda década del siglo XXI fue cuando se empezaron a popularizar los smartphones y tabletas. En un inicio empezaron a competir diferentes sistemas operativos en el mercado como: Symbian, Blackberry OS, Windows Phone, iOS y Android. Sin embargo, actualmente sobreviven estos dos últimos, debido a la preferencia de los usuarios a nivel mundial. De

acuerdo a la empresa Gartner en el año 2016, el 80.7% de los smartphones distribuidos utilizaban el sistema operativo android, mientras que tan solo el 17.7% poseía el sistema operativo iOS. Las estadísticas son claras, y reflejan que el sistema operativo dominante actualmente es Android.

7.5.1 Sistema Operativo “Android”.

Android es un sistema operativo Open Source pensado para teléfonos móviles y desarrollado por Open Handset Alliance (OHA) bajo autorización de Google (Nazim, 2013). Android se basa en un kernel Linux y se distribuye bajo la licencia Apache License 2.0. Este sistema operativo separa la capa de hardware de la capa lógica; esto permite que cualquier teléfono Android pueda ejecutar la misma aplicación, lo cual hace posible que se genere un gran número de posibilidades para los fabricantes, desarrolladores y usuarios.

A pesar de su poco tiempo de vida, éste ha sido uno de los crecimientos más rápidos en lo que ha sistemas operativos se refiere, puesto que a principios de esta década sólo estaba presente en Smartphones, posteriormente también era posible encontrarlo en tabletas y televisores inteligentes. Hoy en día Android está presente en relojes y autos.

Existen diferentes motivos a los que se le deben su éxito. Una de las características más importantes de este sistema operativo reside en que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada (David & Clodoaldo, 2012). Por esta razón es muy popular entre los fabricantes de teléfonos y desarrolladores, visto que los costes para lanzar un teléfono o una aplicación son muy bajos.

Por otra parte, la amplia gama de aplicaciones que están disponibles en su tienda oficial llamada Google Play en donde es posible encontrar un gran número de aplicaciones para satisfacer necesidades específicas del usuario que van desde alarmas y relojes personalizados hasta cursos para aprender un nuevo idioma; además de que Android es muy flexible en cuanto a la personalización, debido a que el usuario tiene la capacidad de modificar e interactuar con el dispositivo Android para adaptarse a los gustos y preferencias del usuario. En la tabla 1 se puede visualizar una comparación entre los dos sistemas operativos para dispositivos móviles que actualmente dominan el mercado.

7.5.2 Comparación entre Android y iOS.

Tabla 1. Comparación entre Android y iOS.

	Android	iOS
Desarrollador	Google	Apple Inc.
Familia de sistema operativo	Linux	OS X, UNIX
Personalización	Mucha, se puede cambiar casi todo	Poca, excepto cuando se desbloquea
Lanzamiento oficial.	23 de septiembre de 2008	29 de Julio de 2007
Programado en	C, C++, Java	C, C++, Objective-C
Dependiente de una PC o Mac	No	No
Código abierto	Núcleo, interfaz y algunas aplicaciones básicas.	El núcleo de iOS no es de código abierto, pero está basado en el código abierto Darwin OS
Disponible para	Muchos celulares y tabletas, incluyendo: Kindle Fire, Samsung, Sony, Motorola, Nexus, y otros.	iPod Touch, iPhone, iPad, Apple TV (2da y 3ra generación)
Tienda de aplicaciones	Google Play Store	Apple App Store

Tabla que muestra las características de los sistemas operativos móviles más populares.

7.6 Lenguaje de programación.

Para que las computadoras y demás dispositivos inteligentes puedan ser útiles para los seres humanos mediante aplicaciones, es necesario que éstas sean programadas mediante un lenguaje de programación. Un lenguaje de programación es un conjunto de reglas y estándares que es utilizado para escribir programas de computador (software), que puedan ser entendidos por él (Joyanes, Rodríguez, & Fernández, Fundamentos de Programación, Libro de problemas., 1996). Es importante considerar que la programación se puede definir como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos (Wilson, 1993).

Hoy en día existen diferentes lenguajes de programación, los cuales son utilizados para programar el software que permite a las personas hacer uso de cualquier computadora. Estos lenguajes han sido creados para satisfacer alguna necesidad informática en específico, por ejemplo, para programar en un entorno web, para desarrollar software de computadoras de escritorio e incluso para desarrolla las aplicaciones que actualmente utilizan los dispositivos móviles. Para desarrollar software dedicado a dispositivos móviles, es importante considerar que prácticamente dos diferentes lenguajes de programación son utilizados. Por el lado de los dispositivos móviles con sistema operativo android el lenguaje java es utilizado para desarrollar sus aplicaciones móviles. Mientras que, por el lado de iOS, Objective-C es el lenguaje más popular en el que se desarrollan las aplicaciones que se encuentran en App Store.

7.6.1 Objective-C.

Como se mencionó anteriormente, Objective-C es el lenguaje de programación más utilizado para desarrollar aplicaciones móviles con sistema operativo iOS. Objective-C, es una extensión del lenguaje C para hacerlo orientado a objetos, pero a diferencia de C++, Objective C está basado en ideas del mundo de Smalltalk, lo cual hace que Objective-C sea un lenguaje más limpio, pequeño y rápido de aprender que C++ (López, 2008).

7.6.2 Java.

Java es un lenguaje de programación creado para desarrollar programas que ejecuten tareas de diversos tipos, y se diseñó de manera concreta para realizar una amplia gama de actividades en internet (Joyanes & Zahonero, Programación en Java, 2011). Sin embargo, hoy en día el lenguaje java es utilizado para desarrollar las aplicaciones que se utilizan en los dispositivos móviles con sistema operativo android.

Java fue elegido como lenguaje de programación para el entorno de desarrollo en android. Haciendo uso del IDE oficial de desarrollo “android studio”. Es importante mencionar que este último, es un potente editor de códigos y ofrece diferentes herramientas para desarrolladores que aumentan la productividad de compilación de apps. A continuación, se describen las características más importantes:

- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- “Instant Run” para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Herramientas “Lint” para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

7.7 Bases de datos.

Las bases de datos son un componente esencial de la vida cotidiana en la sociedad moderna. Actualmente la mayoría de las personas se enfrentan a diversas actividades que implican cierta interacción con una base de datos (Elmasri & Navathe, 2007). Una base de datos es una colección de datos lógicamente coherente con algún tipo de significado inherente, por lo

que no es correcto llamar base de datos a un surtido aleatorio de datos (Elmasri & Navathe, 2007).

Entre los ejemplos más comunes del uso de una base de datos se encuentran: ir al banco a depositar o retirar dinero, realizar una reservación en un hotel de manera online, comprar algún artículo en una tienda de autoservicio, acceder al sistema de una biblioteca para encontrar un libro, o simplemente al interactuar con cualquier tipo de redes sociales. Todas estas actividades implican que la computadora acceda a una base de datos.

7.7.1 Sistemas gestores de bases de datos.

Un Sistema Gestor de Bases de Datos o SGBD se puede definir como una colección de datos relacionada entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD (Ramos, Ramos, & Montero, 2006).

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Como todo sistema, los SGBD se componen diferentes elementos; en este caso, son cuatro:

- a. Lenguajes de los SGBD: Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.
- b. El diccionario de datos: Es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman. Ésta contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.

- c. Seguridad e integridad de datos: Un SGBD proporciona los mecanismos para garantizar la seguridad e integridad de los datos.
- d. El administrador de la base de datos: En los sistemas de gestión de bases de datos actuales existen diferentes categorías de usuarios. Estas categorías se caracterizan por el nivel de privilegios que tiene cada usuario.

Actualmente existen diferentes sistemas gestores de bases de datos en el mercado, entre los más comunes se encuentran: MySQL, dBase, Informix, DB2, PostgreSQL, SQLite, entre otras. Actualmente SQLite es en SGBD más utilizado para almacenar datos en los dispositivos móviles actuales.

7.7.2 SQLite.

Desde la popularización de los dispositivos móviles, el sistema gestor de base de datos llamado SQLite ha sido protagonista de la administración de las bases de datos en dispositivos móviles con sistema operativo Android. SQLite es un paquete de software de dominio público que proporciona un sistema gestor de base de datos o RDBMS (Jay, 2010).

SQLite es una base de datos embebida. En lugar de funcionar independientemente como un proceso autónomo, ésta coexiste simbólicamente en la aplicación a la que sirve. Su código está entrelazado como parte de la aplicación en la que está alojada la base de datos. Para un observador externo nunca sería evidente que ese programa tuviera un sistema gestor de base de datos relacionales a bordo. El programa se limitaría a hacer su trabajo y a administrar los datos de alguna manera sin hacer fanfarrea de cómo se ejecuta en la aplicación (Grant & Mike, 2010).

Una interesante característica de tener un servidor de base de datos dentro del programa o aplicación es que no necesita configuración de red alguna. Esto libera al programador de configuración de firewalls o direcciones o protocolos de resolución de dirección; además de omitir la asignación de privilegios en el sistema. En otras palabras, el programador sólo se preocupa por correr el programa correctamente.

Los sistemas de gestión de base de datos relacionales como SQLite son utilizados para almacenar registros definidos por el usuario en tablas. Además del almacenamiento y

administración de los datos, un motor de base de datos puede procesar comandos de consulta complejos que combinan datos de diferentes tablas con el fin de generar reportes con información significativa para el usuario. En el mercado, es posible encontrar diferentes gestores de base de datos relacionales entre los que se encuentran: Oracle Database, DB2 de IBM, Microsoft SQL Server. Por otra parte, MySQL y PostgreSQL son de código abierto.

Se concibe que SQLite es uno de los gestores de base de datos más versátiles que existen hoy en día. Es una base de datos, una biblioteca de programación y una herramienta de línea de comandos, así como una excelente herramienta de aprendizaje que proporciona una buena introducción a las bases de datos relacionales. Existen diferentes formas de utilizar SQLite: en entornos embebidos, sitios web, servicios de sistemas operativos, scripts y aplicaciones móviles. Para los programadores SQLite es como una cinta conductiva de datos, proporcionando una manera fácil de enlazar las aplicaciones con sus datos (Grant & Mike, 2010).

Al igual que la cinta aislante, no hay fin para sus usos potenciales. En un entorno web, SQLite puede ayudar a administrar información de sesión compleja. Por otra parte, también puede ser utilizado en el desarrollo de aplicaciones móviles en donde las características propias de dicho gestor de base de datos son idóneas para guardar su información.

7.7.2.1 Historia de SQLite.

El autor de SQLite, D. Richard Hipp trabajaba para General Dynamics en un proyecto para el desarrollo de software de la marina estadounidense, dicho programa tenía que ir abordo de misiles guiados de destrucción. El programa funcionaba originalmente en Hewlett-Packard Unix (HP-UX) y utilizaba una base de datos Informix como back-end. Para su aplicación particular, Informix era algo excesivo. Para un administrador de base de datos, en ese momento podría haber tardado casi un día entero en instalar o actualizar a Informix. Entonces para el programador de aplicaciones novato podría tardar aún más. En ese momento lo que realmente se necesitaba era una base de datos autónoma, que fuera fácil de usar, que pudiera viajar con el programa y que pudiera ejecutarse en cualquier lugar sin importar que otro software estuviera instalado en el mismo sistema.

En enero del año 2000, Hipp y un colega discutieron la idea de crear una simple base de datos SQL embebida que usará la librería GNU DBM como back-end que a su vez no necesitará instalación ni soporte administrativo. Más tarde cuando Hipp se desocupó empezó a trabajar en el nuevo gestor de base de datos. Fue así como en agosto del año 2000 se lanzó SQLite 1.0.

Como se había planeado SQLite 1.0 utilizó GDBM como gestor de almacenamiento. Sin embargo, Hipp pronto reemplazó con su propia implementación B-tree, la cual soportaba transacciones y registros almacenados en orden de claves. Con la primera actualización importante en la mano, SQLite empezó una nueva evolución constante crecimiento en características y usuarios. A mediados de 2001 muchos proyectos, tanto de código abierto, como comerciales comenzaron a utilizarlo. En los años que siguieron, otros miembros de la comunidad de código abierto comenzaron a escribir extensiones de SQLite para sus lenguajes favoritos de programación, además de librerías. Una por una, surgieron nuevas extensiones para lenguajes de programación populares como Perl, Python, Ruby, Java, entre otros.

SQLite comenzó una actualización importante de la versión 2.0 a la 3.0 en el año 2004. Su principal objetivo era la internacionalización mejorada que soporta el texto UTF-8 y UTF-16, así como secuencias de texto definidas por el usuario. Aunque la versión 3.0 estaba programada para su lanzamiento en el verano del 2005, America Online proporcionó los fondos necesarios para verla terminada en Julio de 2004. Además de la internacionalización, la versión 3.0 trajo muchas nuevas características como un formato más compacto para archivos de base de datos y una API mejorada.

SQLite continúa creciendo en función de las características, pero sigue siendo fiel a sus objetivos iniciales de: diseño, simplicidad, flexibilidad, compacidad, velocidad y facilidad de uso general (Grant & Mike, 2010).

7.7.2.2 SQLite en Android.

En programación, a medida que aumenta el número de datos en el desarrollo de una aplicación, es necesario almacenarlos en una base de datos apropiada para su posterior uso.

Esta práctica permite optimizar los recursos propios del dispositivo además del tiempo en consultar determinada información.

SQLite es el gestor de base de datos que se adapta mejor al desarrollo de aplicaciones móviles en Android. SQLite ha ganado mucha popularidad en los últimos años por ofrecer características tan interesantes como su poca configuración, no necesita de un servidor para funcionar, su escaso tamaño y es de código libre.

SQLite se encuentra como una librería para ser utilizada en el IDE de desarrollo Android Studio, a la cual se puede acceder a través de la clase SQLiteOpenHelper. Para poder hacer uso de este gestor de base de datos en un proyecto en Android Studio es necesario crear una clase que se extienda a SQLiteOpenHelper. En la figura 1 se puede observar la sintaxis de la clase necesaria para poder utilizar SQLite en android studio.

```
public class DatabaseHelper extends SQLiteOpenHelper {  
  
}
```

Figura 1 Clase que extiende a SQLiteOpenHelper.

Para el presente proyecto se utilizará SQLite como gestor de base de datos puesto que posee las características adecuadas para realizar las diferentes consultas que se requieren.

En el caso del IDE Android Studio será necesario crear dos métodos importantes en la clase SQLiteOpenHelper. Estos dos métodos son: onCreate y onUpgrade. El primero contendrá la sintaxis en lenguaje SQL para crear las tablas de la aplicación a desarrollar mientras que el segundo método tendrá los parámetros necesarios para hacer posteriores modificaciones a la base de datos.

7.7.3 Modelo relacional.

Para poder hacer uso de la estructura lógica de una base de datos es necesario representarla mediante un modelo. El modelo más utilizado para hacer este tipo de representaciones es el

modelo relacional. En otras palabras, un modelo relacional permite representar la forma en que una base de datos está construida, además de cómo sus datos interactúan entre sí. El modelo relacional representa la base de datos como una colección de relaciones. Informalmente, cada una de estas relaciones se parece a una tabla de valores o, de forma algo más extensa, a un fichero plano de registros (Ramez & Shamkant, 2007).

7.7.4 Modelo entidad relación.

Para modelar una base de datos, es ideal contar con un diagrama que represente las entidades de una base de datos, así como sus interrelaciones de esta. Este modelo es llamado de “entidad-relación”. Dicho modelo intenta describir “el mundo real” como un conjunto de entidades y relaciones entre ellas. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos (Ramez & Shamkant, 2007).

7.7.5 Modelo entidad relación extendida.

Como se ha descrito anteriormente, un modelo entidad relación, intenta describir el mundo real mediante entidades y las relaciones que existen entre ellas, sin embargo para efectos de programación es importante adaptar este modelo a uno que se parezca más a la programación orientada a objetos, Este modelo representa de forma más precisa los requerimientos de una base de datos (Mannino, 2007).

7.7.6 Normalización de bases de datos.

La normalización es un proceso de eliminación de redundancias en una tabla para que sea más fácil de modificar (Mannino, 2007). A continuación, se muestran las tres formas normales para hacer que una base de datos funcione óptimamente.

7.7.6.1 Primera forma normal.

La primera forma normal ayuda a no repetir datos en las tablas que componen la base de datos.

7.7.6.2 Segunda forma normal.

LA segunda forma normal intenta que cada columna de la tabla debe depender de la clave, esto significa que cada registro debe depender únicamente de la clave principal.

7.7.6.3 Tercera forma normal.

En cuanto a la tercera forma normal es necesario asegurarse de que ninguna columna dependa de una columna que no tenga una clave.

7.8 Inteligencia Artificial (IA).

Los seres humanos se han autodenominado “Homo Sapiens” (Hombre Sabio) debido a que sus capacidades mentales son muy importantes, y en cierta forma, la inteligencia es lo que nos separa del resto de los animales con quienes convivimos en este planeta. De hecho, la inteligencia es lo que ha hecho que el hombre intente comprender, predecir y manipular el mundo que lo rodea. Debido a los avances tecnológicos del siglo pasado, el hombre ha tratado de emular su inteligencia, intentando crear máquinas que sean capaces de remplazarlo e incluso superar sus capacidades intelectuales. Es aquí donde surge una de las ciencias más recientes como lo es la Inteligencia artificial (IA).

La inteligencia artificial puede ser concebida como la automatización de actividades que se vinculan con procesos de pensamiento humano, actividades tales como la toma de decisiones, resolución de problemas, aprendizaje, etcétera (Bellman, 1978). Por otra parte, la IA también puede ser vista como el arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren inteligencia (Kurzwil, 1990). La IA tiene sus orígenes en 1950 con el trabajo de Alan Turing, quien fue un matemático británico que diseñó el primer computador electrónico digital y funcional del mundo. La máquina de Turing fue un modelo

computacional en el cual se estudiaba la cuestión planteada por David Hilbert sobre si las matemáticas son decidibles, es decir, si hay un método definido que pueda aplicarse a cualquier sentencia matemática y que ésta diga si esa sentencia es cierta o no. Turing construyó un modelo formal de computador, la máquina de Turing, y demostró que existían problemas que una máquina no podía resolver. La máquina de Turing es un modelo matemático abstracto que formaliza el concepto de algoritmo (Kurzweil, 1990). Posteriormente, empezaron a haber más avances en cuanto al estudio de la IA, al grado de hacer de ésta una ciencia que hoy en día tiene una gran importancia en la vida de los humanos. Al igual que otras ciencias, la IA se puede dividir actualmente en cinco campos diferentes como lo son:

- **Sistemas expertos:** Programas computacionales que resuelven problemas que normalmente requieren del conocimiento de un especialista o experto humano. Es un sistema capaz de tomar decisiones inteligentes interpretando grandes cantidades de datos sobre un dominio específico de problemas.
- **Aprendizaje y razonamiento automático:** Máquinas capaces de planificar, tomar decisiones, plantear y evaluar estrategias, aprender a partir de la experiencia, autoreprogramables, etc.
- **Robótica:** Artefactos autónomos capaces de llevar a cabo diversas tareas mecánicas de manera flexible e inteligente, cumpliendo con un objetivo y ajustándose al entorno cambiante.
- **Visión por computadora (reconocimiento de patrones):** Reconoce y procesa señales, caracteres, patrones, objetos, escenas.
- **Procesamiento de lenguaje natural:** Sistemas capaces de reconocer, procesar y emular el lenguaje humano.

Este último campo es el encargado estudiar todo lo relacionado al reconocimiento automático de voz, que actualmente es utilizado para controlar dispositivos móviles por medio de comandos de voz. Haciendo la interacción entre humano – computadora más fácil.

7.9 Reconocimiento automático de voz.

El Reconocimiento Automático del Habla (RAH) o Reconocimiento Automático de Voz es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas (Reyes Bascuñana & Satofimia Ruiz).

7.9.1 Historia de los sistemas RAH.

Los primeros intentos por construir máquinas que realizaran tareas de reconocimiento del habla se remontan a la década de los 50, cuando diversos investigadores trataban de explotar los principios fundamentales de la fonética acústica (Reyes Bascuñana & Satofimia Ruiz).

En el año de 1952, los laboratorios Bell hicieron un gran avance creando un sistema electrónico que permitía identificar para un solo hablante pronunciaciones de los 10 dígitos numéricos realizadas de forma aislada. Esta máquina se basaba en medidas de las resonancias espectrales del tracto vocal para cada dígito.

Más tarde en los años 60 se comenzó a experimentar con normalización temporal según la detección de los puntos de comienzo y fin de las palabras.

En los años 70 hubo más interés por parte de los investigadores en esta área. A causa de que las investigaciones se centraron en el reconocimiento de palabras aisladas. También comenzaron los primeros intentos de reconocimiento de habla continua y la experimentación en reconocimiento independiente del locutor.

Posteriormente en los años 80 se produce un giro metodológico como consecuencia de pasar de métodos basados en comparación de plantillas a los métodos basados en modelado estadístico (Reyes Bascuñana & Satofimia Ruiz). Se empezaron a utilizar los modelos ocultos de Markov (HMM), obteniendo excelentes resultados en el modelado de voz. A partir de entonces se han desarrollado numerosas mejoras y actualmente constituyen los mejores modelos disponibles para capturar y modelar la variabilidad del habla.

En la última década del siglo pasado se les dio continuidad a los objetivos propuestos anteriormente ampliando el tamaño de los vocabularios a la vez que se diversifican los campos de aplicación.

En los inicios del siglo XXI, las investigaciones relacionadas al reconocimiento automático de voz se han enfocado en aplicar dichas tecnologías para solucionar problemas particulares de las personas, es por ello que la mayoría de smartphones cuentan con esta tecnología para brindarles a los usuarios una nueva interfaz para comunicarse.

Desde que Apple en el año 2011 hizo el lanzamiento de Siri junto con el iPhone 4s, fue cuando realmente se le puso atención al reconocimiento de voz por parte de los usuarios y fabricantes de teléfonos inteligentes. Aunque esta tecnología lleva funcionando varias décadas, en los últimos años se le ha tenido más aplicaciones en la vida diaria de las personas que hacen uso de smartphones y computadoras.

Si bien es cierto que el reconocimiento de voz desarrollado en 1952 por los laboratorios Bell, tenía algunas deficiencias al momento de reconocer la voz humana, con el paso del tiempo esta tecnología ha mejorado notablemente, al grado de que hoy en día el reconocimiento de voz tiene un gran nivel de precisión que es difícil diferenciar la ciencia ficción de la realidad. Este mejoramiento se debe en gran medida a los avances tecnológicos en el hardware y software.

7.9.2 Caracterización de los sistemas de reconocimiento de voz.

Los sistemas de reconocimiento de voz se pueden clasificar tomando en cuenta diferentes aspectos:

1. La forma en que el usuario le habla a la máquina. Existen básicamente tres formas:
 - a. Palabra aislada: El usuario habla palabras individuales (o frases) tomadas de un vocabulario determinado.
 - b. Palabras conectadas: El usuario habla en forma fluida una sucesión de palabras pertenecientes a un vocabulario restringido (ej. dígitos telefónicos).
 - c. Habla continua: El usuario habla fluidamente usando palabras de vocabulario grande (usualmente ilimitado).

2. Tamaño del vocabulario de reconocimiento.
 - a. Pequeño: Capaz de reconocer hasta 100 palabras.
 - b. Mediano: Entre 100 y 1000 palabras.
 - c. Grande: Más de 1000 palabras.
3. El conocimiento de los patrones de voz del usuario.
 - a. Sistemas dependientes del locutor: Adaptados a locutores particulares.
 - b. Sistemas independientes de locutor: Trabajan con una población de locutores grande, la mayoría de los cuales es desconocidos para el sistema.
 - c. Sistemas adaptables: Se adaptan al locutor particular mientras el sistema está en uso.
4. Grado de conocimiento acústico-lingüístico usado por el sistema.
 - a. Sólo conocimiento acústico. No usan conocimiento lingüístico.
 - b. Integración de conocimiento acústico y lingüístico. El conocimiento lingüístico esta usualmente representado por restricciones sintácticas y semánticas sobre la salida del sistema de reconocimiento.
5. Grado de dialogo entre el usuario y la máquina.
 - a. Unidireccional (o pasivo): El usuario habla y la máquina realiza una acción como respuesta.
 - b. Sistema de dialogo activado por la máquina. El sistema es el iniciador del diálogo, requiriendo información del usuario vía una entrada verbal.
 - c. Sistema de diálogo natural. La máquina conversa con el locutor, le solicita entradas, actúa en función de las entradas y trata de clarificar ambigüedades.

7.9.3 Fuentes de variabilidad de las señales de voz.

El reconocimiento automático de voz es una tarea inherente difícil debido a la variabilidad de las señales de voz. Algunas fuentes de variabilidad incluyen:

1. Variabilidad en el locutor en mantener una pronunciación consistente en el uso de palabras y frases.
2. Variabilidad entre locutores debido a diferencias fisiológicas (ej. Diferente longitud de tracto vocal), acentos regionales, idiomas extranjeros, etc.

3. Variabilidad entre transductores cuando se habla frente a diferentes micrófonos o aparatos telefónicos.
4. Variabilidad introducida por el sistema de transmisión (redes de comunicación, teléfonos celulares, etc.).
5. Variabilidad en el ambiente, que incluyen conversaciones extrañas y eventos acústicos de fondo, como ruidos, etc.

7.9.4 Reconocimiento de voz en la nube de Google.

Google cuenta con una API de reconocimiento de voz en la nube, cabe mencionar que cuenta con varias funcionalidades interesantes cuando se necesita una interfaz de desarrollo de aplicaciones vinculada al procesamiento de lenguaje natural, reconocimiento de voz y obtención de resultados en tiempo real (Google, 2018). Cuenta con las siguientes funcionalidades:

- **Reconocimiento de voz automático (Automatic Speech Recognition - ASR):** usa una red neuronal de aprendizaje profundo para el reconocimiento de voz, aportar funcionalidades de búsqueda por habla y su transcripción.
- **Reconocimiento en streaming:** a medida que la API va procesando y reconociendo la voz del usuario va devolviendo resultados en tiempo real, sin tiempos de espera. Es lo que facilita que la aplicación pueda ofrecer al cliente todas las funcionalidades en torno a ese procesamiento.
- **Soporte búfer de audio:** la API es capaz de procesar el sonido procedente del micrófono de una aplicación o del dispositivo móvil y empaquetarlo en distintos tipos de formatos de compresión: FLAC, AMR, PCMU y linear-16. Esa compresión es necesaria para el tratamiento posterior del sonido.
- **Reconocimiento de voz en más de 80 idiomas distintos.** Esta es una característica que ofrece una gran ventaja competitiva con otros proveedores de servicios similares a terceros desarrolladores.

7.9.4.1 Peticiones de voz.

A continuación se describen los tipos de solicitudes que puede realizar Speech-to-Text, cómo crear esas solicitudes y cómo manejar sus respuestas.

Speech-to-Text tiene tres métodos principales para realizar reconocimiento de voz (Google, 2018). Estos se enlistan a continuación:

- **El reconocimiento síncrono** (REST y gRPC) envía datos de audio a la API de voz a texto, realiza el reconocimiento de esos datos y devuelve los resultados una vez que se ha procesado todo el audio. Las solicitudes de reconocimiento síncrono están limitadas a datos de audio de 1 minuto o menos de duración.
- **El reconocimiento asíncrono** (REST y gRPC) envía datos de audio a la API de voz a texto e inicia una operación de ejecución prolongada. Usando esta operación, puede sondear periódicamente para obtener resultados de reconocimiento. Use solicitudes asíncronas de datos de audio de cualquier duración hasta 180 minutos.
- **El Reconocimiento de Streaming** (solo gRPC) realiza el reconocimiento en datos de audio proporcionados dentro de una transmisión bidireccional de gRPC. Las solicitudes de transmisión están diseñadas para fines de reconocimiento en tiempo real, como la captura de audio en vivo desde un micrófono. El reconocimiento de transmisión proporciona resultados provisionales mientras se captura el audio, permitiendo que aparezca el resultado, por ejemplo, mientras el usuario sigue hablando.

7.9.4.2 Reconocimiento de voz utilizando redes neuronales

El CSLU toolkit, desarrollado por el Center for Spoken Language Understanding (CSLU) del Oregon Graduate Institute (OGI), es un ambiente de software para investigación y desarrollo, que provee una herramienta poderosa y flexible para crear y usar sistemas de lenguaje hablado. Uno de los ambientes de desarrollo de reconocedores es CSLU-NN, una herramienta que nos permite crear reconocedores de voz basados en redes.

Los reconocedores son desarrollados usando muestras de la señal tomadas cada t unidades de tiempo, lo que llamamos marcos o frames, junto con una red neuronal para estimar las probabilidades posteriormente. Los pasos realizados durante el reconocimiento se muestran en la figura 2 y son los siguientes:

- Dividir la señal en muestras (marcos);
- Extraer las características de cada marco. Estas características describen el espectro que envuelve a la señal en ese marco y en un pequeño número de marcos vecinos. Las características de cada muestra son clasificadas en categorías fonéticas usando una red neuronal. Las salidas de la red neuronal son usadas como estimaciones de probabilidad para cada categoría fonética de la muestra actual.
- Determinar la(s) palabra(s) más parecidas a través de la búsqueda Viterbi. Esta búsqueda usa una matriz de probabilidades (las salidas de la red neuronal) y un conjunto de modelos de pronunciaciones.

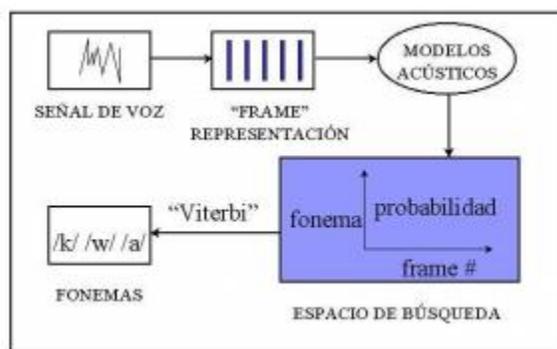


Figura 2. Sistemas de reconocimiento de voz basado en redes neuronales.

7.9.5 Modelos ocultos de Markov para el reconocimiento automático del habla.

Los modelos ocultos de Markov (MOM) constituyen una de las técnicas que se ha utilizado con más éxito en el reconocimiento automático del habla (RAH). Principalmente, esta técnica ha permitido modelar adecuadamente la gran variabilidad en el tiempo de la señal de voz. En la terminología del RAH, con MOM suele hacerse referencia no solo a la técnica de los modelos ocultos de Markov propiamente dicha, sino también a una larga lista de adaptaciones

y técnicas asociadas que se fueron incorporando para solucionar el problema de RAH (Oropeza & Suárez, 2006). En la figura 3 se muestra un ejemplo de transición de estados en un modelo oculto de Markov, en donde “x” representa los estados ocultos, “y” las salidas observables, “a” las probabilidades de transición y “b” las probabilidades de salida.

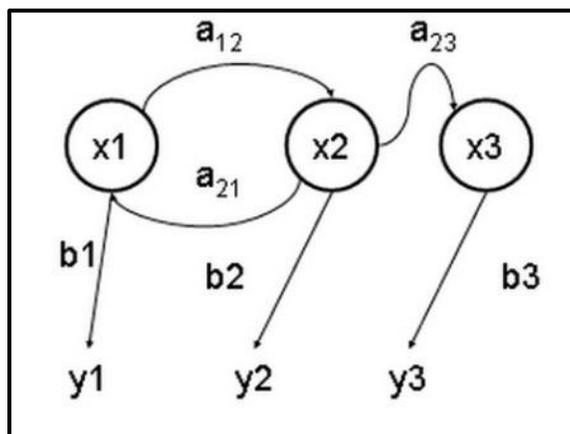


Figura 3. Ejemplo de transición de estados en un modelo oculto de Márkov.

7.10 Lenguaje Formal

Para el desarrollo de presente proyecto fue imperativo hacer uso de una sintaxis genérica para que el usuario se comunique con la app por medio del reconocimiento automático de voz. En este sentido se utilizó un lenguaje formal para poder describir correctamente la sintaxis a utilizar. En diferentes disciplinas tales como las matemáticas, la lógica y las ciencias de la computación un lenguaje formal se constituye por símbolos primitivos (alfabeto) y reglas para unir dichos símbolos (gramática formal) (Chomsky, 1956).

7.10.1 Analizador Sintáctico.

Como se ha mencionado anteriormente, se tomó en cuenta el uso de una gramática formal, la cual proporciona reglas de sintácticas para que el usuario pueda comunicarse con la app. En el área de la informática, un analizador sintáctico puede ser definido como un programa encargado de analizar una cadena de símbolos de acuerdo a las reglas de una gramática formal (Grune & Jacobs, 2008).

El analizador sintáctico convierte al texto de entrada en otras estructuras para su posterior análisis. En este caso, la API de reconocimiento automático de voz se encarga de convertir la voz en cadenas de texto para que posteriormente se pueda hacer uso de dichas cadenas.

7.10.2 Analizador Lexicográfico.

Un analizador lexicográfico es un programa que recibe como entrada una secuencia de caracteres y a su vez, este produce una salida compuesta por tokens o símbolos (Ravi Sethi & Ullman, 1986) en este caso, la secuencia de caracteres es la cadena de texto que interprete la API de reconocimiento automático de voz y los tokens serán las palabras de las que se componga dicha cadena. Es importante en análisis lexicográfico debido a que la app reconocerá la suma de tokens para interpretar la sintaxis ingresada. Si la sintaxis es correcta, la app procederá a hacer las instrucciones correspondientes.

7.10.3 Tokenización.

La tokenización es el proceso en donde un programa informático convierte una cadena de texto ingresada a una cadena o cadenas de texto con un significado coherente en los lenguajes de programación. Para el presente proyecto de investigación los tokens se pueden conceptualizar como las palabras que compongan las oraciones interpretadas por el reconocimiento automático de voz.

7.11 Autómatas.

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos, es decir, de las “máquinas”. Antes de que existieran las computadoras, en la década de los años treinta, A. Turing estudió una máquina abstracta que tenía todas las capacidades de las computadoras de hoy día, al menos en lo que respecta a lo que podían calcular. El objetivo de Turing era describir de forma precisa los límites entre lo que una máquina de cálculo podía y no podía hacer; estas conclusiones no sólo se aplican a las máquinas abstractas de Turing, sino a todas las máquinas reales actuales (Hopcroft, Motwani, & Ullman, 2008).

7.11.1 Diagrama de transiciones.

Con el objetivo de desarrollar una aplicación estandarizada se ha optado por hacer uso de un diagrama de transiciones. Este diagrama sirve para mostrar el comportamiento de un sistema en cuanto a los diferentes estados que puede tomar éste, en un periodo de tiempo determinado. Los diagramas de estados son especialmente importantes para describir el comportamiento de un sistema reactivo, cuyo comportamiento está dirigido por eventos (Booch, Rumbaugh, & Jacobson, 1999).

Un diagrama de transiciones esta básicamente formado por dos elementos: “estado” y transición.

Un estado, representa algún comportamiento que es observable en el sistema y que perdura durante un periodo de tiempo finito. Viene dado por el valor de uno o varios atributos que lo caracteriza en un momento dado. El estado o estados son representados por un circulo y en el centro de este tiene escrito una letra “qn”. Cabe mencionar que en el primer estado (q0) esta apuntado por una flecha, la cual indica que desde allí comienza el diagrama. Cada uno de los estados finales está representado por un circulo doble, el cual indica que en este finaliza algún proceso.

Una transición, se puede concebir como un cambio de estado producido por un evento, y refleja los posibles caminos para llegar a un estado final desde un estado inicial. Cada una de las transiciones está representada por una flecha que proviene de un estado y apunta a otro estado.

7.11.2 Tabla de transiciones.

Una tabla de transiciones se puede concebir como una matriz cuyos elementos proporcionan el resumen de un diagrama de transiciones correspondiente. En el presente proyecto se utilizará una tabla de transiciones que también se le denominará como tabla sintáctica. Dado que en ella se formalizarán las cadenas de texto que se ingresarán a la aplicación por medio de comandos de voz. Todo lo anterior con el fin de crear una serie de instrucciones por voz que obedezcan a las diferentes tareas que llevará a cabo la aplicación.

7.12 Dispositivos móviles.

Un dispositivo móvil es un aparato electrónico que permite el procesamiento de datos mediante las aplicaciones que pueden ser instaladas en él, con el fin de cubrir alguna necesidad del usuario. Gracias a su pequeño tamaño y escaso peso son fáciles de transportar incluso en un bolsillo.

Debido a la fuerte competencia de este sector los cambios en estos dispositivos, su variedad y características evolucionan rápidamente. Sin embargo, se mantienen una serie de características comunes. La mayoría cuentan con una pantalla generalmente táctil y un teclado que hacen posible la entrada y salida de datos.

7.13 Aplicaciones Móviles.

Las aplicaciones móviles se pueden concebir como el software o programas que poseen los dispositivos móviles con la finalidad de satisfacer una o varias necesidades de un usuario. Las funciones de cada aplicación disponible van en función de las características de los dispositivos y sistema operativo de los dispositivos que se pretenda instalar. Estas características van desde el procesador, memoria RAM e incluso capacidad de almacenamiento.

7.14 Metodologías de desarrollo ágil.

Durante el año 2001 nació el término “ágil” aplicado al área de desarrollo de software, la esencia de las metodologías ágiles se basa en valores y principios que deberían permitir a los equipos desarrollar software rápidamente y responder a los cambios que pueden surgir a lo largo del proyecto. Este tipo de metodologías se pueden concebir como una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas.

Las metodologías de desarrollo ágiles se basan en los siguientes principios (Letelier, 2006):

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también puede afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- **Desarrollar software que funciona más que conseguir una buena documentación.** Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.
- **La colaboración con el cliente más que la negociación de un contrato.** Las características particulares del desarrollo de software hacen que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso

del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

7.14.1 Metodología Crystal.

La metodología Crystal es un enfoque ágil de desarrollo de software que se enfoca principalmente en sus personas y sus interacciones cuando se trabaja en un proyecto en lugar de en procesos y herramientas. Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos (Pantaleo & Ludmila, 2014).

Esta metodología se basa en dos principios.

- Los equipos pueden optimizar sus procesos como su trabajo y convertirse en un equipo más optimizado.
- Los proyectos son únicos y dinámicos y requieren métodos específicos.

Dado que esta metodología es ágil, tiene las siguientes propiedades que la hacen muy adaptativa a los cambios y resalta la colaboración del equipo de trabajo.

Entrega frecuente: le permite entregar con frecuencia código probado y en funcionamiento a usuarios reales. De esta forma, no tiene que enfrentar el hecho de que ha invertido su energía y tiempo en el producto que nadie quiere comprar.

Mejoramiento reflexivo: no importa cuán malo o bueno sea el producto, siempre hay áreas donde se puede mejorar el producto. Además, siempre hay nuevas técnicas y métodos que su equipo puede implementar para mejorar sus prácticas futuras.

Comunicación osmótica: con el equipo que trabaja compartido, la información fluye alrededor del equipo. Eso les permite recoger información valiosa sin siquiera estar directamente involucrados en la discusión de la cuestión determinada. Esta absorción gradual de ideas se denomina comunicación osmótica.

Seguridad personal: la única forma de crear un ambiente de trabajo saludable y una verdadera cultura de equipo es practicando una comunicación abierta y honesta. Los miembros del equipo deben poder hablar sin miedo, sin importar si presentan una nueva idea o si hablan sobre un posible problema.

Enfoque: cada miembro del equipo sabe exactamente en qué trabajar, lo que les permite concentrar su atención y evitar el cambio de una tarea a otra. Además, esto aumenta la comunicación del equipo y ayuda al equipo a priorizar y trabajar para lograr los mismos objetivos.

Acceso fácil a usuarios expertos: Crystal permite a su equipo mantener la comunicación y recibir comentarios frecuentes de usuarios reales.

Un entorno técnico con pruebas automatizadas, gestión de configuración e integración frecuente: herramientas muy específicas para equipos de software en las que se hace hincapié en la integración continua para que los errores puedan detectarse en cuestión de minutos.

7.14.2 Scrum.

Actualmente, debido a los cambios tan rápidos que surgen en el ámbito empresarial y tecnológico, es necesario implementar nuevas metodologías que faciliten el lanzamiento de productos orientados a una entrega temprana y que a la vez proporcione resultados tangibles. En el ámbito de desarrollo de software existe una metodología ágil que intenta dar solución a los nuevos retos que se presentan hoy en día en cuanto a la evolución rápida de las necesidades de los clientes. Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible en un proyecto (Pressman, Ingeniería del Software, un enfoque práctico, 2010).

Un proyecto en Scrum se ejecuta en bloques temporales, cortos y fijos también llamados iteraciones. Cada iteración tiene que proporcionar un resultado completo, un incremento del producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente que lo

solicite. En la figura 4 se puede apreciar un diagrama que ejemplifica de una forma simple los pasos que conforman la metodología Scrum.

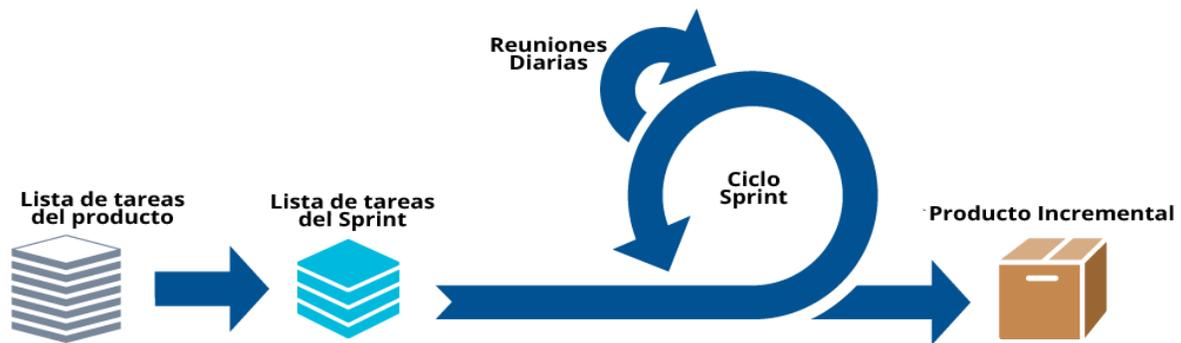


Figura 4. Diagrama de la metodología Scrum.

7.13.2.1 Proceso.

El proceso de desarrollo de la metodología Scrum se ejecuta en bloques temporales cortos y fijos (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite (Proyectos Agiles, 2017).

7.13.2.2 Sprint.

El corazón de Scrum es el Sprint, es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo. Es importante considerar que, durante el sprint, no se realizan cambios que puedan afectar al objetivo del sprint y los objetivos de calidad no disminuyen.

7.13.2.3 El equipo Scrum.

Para hacer uso de la metodología Scrum es necesario conocer de los elementos que la forman. Consiste en: el dueño del producto (product owner), el equipo de desarrollo, Scrum master y stakeholder.

7.13.2.3.1 Dueño del producto (Product owner).

El dueño del producto es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. En otras palabras, es el líder en cuanto a la toma de decisiones del desarrollo del producto. Él es la única autoridad responsable de decidir qué características y funcionalidades se deben construir y el orden en el cual estas deben ser construidas (Kenneth, 2012).

7.13.2.3.2 Equipo de desarrollo (Team).

El equipo de desarrollo está formado por diferentes personas, como el diseñador, programador, administrador de bases de datos, etcétera. Se puede decir que es el conjunto de personas que desarrollan de manera conjunta el proyecto. Estas personas comparten la responsabilidad que realizan en cada iteración y en el proyecto en general (Kenneth, 2012).

7.13.2.3.3 Scrum Master.

El Scrum Master es el responsable de que todos los involucrados entiendan y adopten los valores, principios y prácticas de Scrum. También el Scrum master actúa como un entrenador proporcionando liderazgo y ayudando al equipo de Scrum y al resto de la organización a desarrollar el modelo Scrum propio de la organización. Por otra parte, es responsable de proteger al equipo de interferencia externa y toma el liderazgo al momento de liberar los impedimentos que disminuyen la productividad del equipo (Kenneth, 2012).

7.13.2.3.4 Stakeholder.

Son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del "sprint" (Kenneth, 2012).

7.15 Patrones de diseño.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. En si los patrones de diseño son una forma reutilizable de un problema en común.

7.15.1 MVC

El patrón de diseño MVC es uno de los patrones de diseño más importantes en ciencias de la computación. Mientras que la mayoría de los patrones abordan problemas específicos, éste describe la arquitectura de un sistema de objetos. Se puede aplicar a subsistemas aislados o aplicaciones completas. El patrón de diseño MVC también está menos definido que muchos otros patrones, lo que deja mucha libertad para implementaciones alternativas. Es más, una filosofía que una receta (Bucanek, 2009). A continuación, se describen cada uno de los elementos que componen al modelo MVC:

- **El Modelo**, contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- **El Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

7.15.2 DAO.

La mayoría de las aplicaciones persisten datos y para esto es necesario la aplicación interactúe con la base de datos. El cómo lo hace no debe ser asunto de la capa de lógica de negocio, ya que para eso está la capa de persistencia que es la que hará el intercambio de información con la base de datos. Sabiendo esto, se puede decir que DAO es un patrón de diseño utilizado para crear esta capa de persistencia.

DAO encapsula el acceso a la base de datos. Por lo que cuando la capa de lógica de negocio necesite interactuar con la base de datos, va a hacerlo a través de la API que le ofrece DAO.

Generalmente esta API consiste en métodos CRUD (Create, Read, Update y Delete) (Pressman, Ingeniería del software, Un enfoque práctico, 2010).

7.15.3 DTO.

El Modelo DTO es utilizado por el modelo DAO para transportar los datos desde la base de datos hacia la capa de lógica de negocio y viceversa. Por ejemplo, cuando la capa de lógica de negocio llama al método create(), en DAO inserta un nuevo dato este dato es el que la capa de lógica de negocio le pase como parámetro y este dato se pasa a través de un DTO.

Un DTO es un objeto común y corriente, que tiene como atributos los datos del modelo, con sus correspondientes accessors (getters y setters) (Pantaleo & Ludmila, 2014).

7.15.4 Capa de acceso a datos (DAL).

Se puede entender que la capa de acceso a datos en un programa informático es una capa que proporciona acceso simplificado a los datos almacenados en el almacenamiento persistente de algún tipo, tal como una entidad – relación de una base de datos (Pantaleo & Ludmila, 2014).

7.15.5 Lógica de negocio (Business Layer).

El modelo MVC tiene la particularidad de dividir la complejidad de un proyecto de software en capas independientes con fines prácticos. En este sentido, la lógica de negocio o también conocida como Business Layer es la parte del programa que codifica las reglas de negocio del mundo real que determinan como se pueden crear, almacenar y modificar los datos (McLaughlin, 2002). Para desarrollar el presente proyecto se creó una capa de negocios, que es básicamente una clase que se encarga de enlistar los métodos utilizados en la app y que a su vez corresponden a las reglas de negocio solicitadas.

VIII METODOLOGÍA

La ingeniería de software se ha aplicado al desarrollo de software desde hace aproximadamente treinta años, lo cual ha contribuido directamente a mejorar la calidad de los productos finales y a garantizar el cumplimiento de las necesidades de los usuarios del software.

En la última década los dispositivos móviles se han popularizado en diferentes ámbitos sociales, económicos y políticos. Este tipo de dispositivos han hecho posible la construcción de aplicaciones de gran alcance, adaptables a cambios de contexto, como las variaciones en el ancho de banda, batería, conectividad, accesibilidad de los servicios, diseño de interfaces, capacidad de memoria, entre muchos otros aspectos.

Cuando se plantea desarrollar una aplicación para un dispositivo móvil en cualquiera de las plataformas y para cualquier entorno, es importante reconocer y establecer condiciones que garanticen la funcionalidad, calidad, seguridad y eficiencia de la aplicación que se desea construir y utilizar por medio del dispositivo móvil. Por tal motivo, es de suma importancia seguir en forma clara las etapas generales del ciclo de vida como lo son:

- Análisis.
- Diseño.
- Codificación.
- Pruebas.
- Mantenimiento.

Sin embargo, es necesario considerar las grandes diferencias que existen entre el desarrollo de una aplicación para ejecutar en una computadora de escritorio y una aplicación para ejecutar en un dispositivo móvil. Esto con el fin de desarrollar software que garantice la movilidad, el fácil uso y el aprovechamiento de los recursos limitados de memoria y procesamiento.

Para dicho desarrollo, es necesario adoptar un modelo que estandarice las diferentes etapas del desarrollo del software, además de una metodología que por medio de buenas prácticas regule el ciclo de vida con el fin de obtener un producto de calidad.

Es por ello que, de acuerdo a la problemática planteada es importante utilizar una metodología ágil para el desarrollo del presente proyecto. En este caso la metodología que se utilizará será Scrum ágil. A continuación, se describe de manera detallada el proceso de desarrollo de la aplicación móvil:

8.1 Descripción del negocio (caso práctico).

El desarrollo del presente trabajo de tesis fue basado en un negocio de venta de alimentos y bebidas. Dicho negocio es una taquería llamada “Taquería Espinoza”, dicho negocio es propiedad del C. Adelaido Espinoza Cortes quien en el presente proyecto tomó el papel de cliente. La taquería está ubicada calle Degollado s/n, Texcoco de Mora Edo. Mex.

Como la mayoría de PYMES que pertenecen al sector de venta de alimentos y bebidas, la Taquería Espinoza no hacía uso de algún sistema informático que la apoyara a llevar el control de ventas. En cuanto al número de personal, esta taquería cuenta con dos trabajadores. El primero cumple las labores de cocinero, mientras que el segundo funge como mesero. En su lista de productos existen tres diferentes productos respecto al precio: tacos de \$8.00, tacos de \$12.00 y bebidas embotelladas de \$13.00. En cuanto al número de mesas disponibles son cinco. Sin embargo, cabe mencionar que en momentos donde la clientela es fluida había clientes que no ocupaban mesa debido a que todas ya habían sido utilizadas, pero aún así cada cliente representa un pedido por separado.

8.2 Problemática del negocio.

Como se mencionó en la descripción del negocio, todos los procesos que se llevaban a cabo mediante métodos manuales, en ningún proceso se hacía uso de algún tipo de sistema informático. Esto generaba principalmente un problema recurrente en cuanto a los errores del cobro de las cuentas generadas por cada uno de los clientes. El método que se utilizaba en esta empresa era generar la cuenta total de cada cliente de forma manual. De acuerdo con la versión del propietario, con el paso del tiempo el mesero que era quien generaba las cuentas adquiría la habilidad de reducir el tiempo en generar la cuenta y minimizar el margen de error. Sin embargo, en horarios pico era muy frecuente que el mesero cometiera errores en

cuanto a la exactitud de las cuentas. En ocasiones se les cobraba de más o de menos a los clientes. La consecuencia del primer caso era que el cliente se molestara porque le cobraban de más y en ocasiones de iba con la competencia. En el segundo caso, la empresa tenía pérdidas debido a que se le cobraba de menos al cliente.

8.3 Requisitos generales.

Como se ha mencionado anteriormente en el marco teórico, el sistema operativo más popular en dispositivos móviles es Android, ya que este abarca más de la mitad del mercado. Es por ello que se la app está orientada para dispositivos Android. En este sentido, el lenguaje en el que fue desarrollada la app fue Java, ya que la programación fue haciendo uso del IDE de desarrollo Android Studio.

Actualmente existen varias versiones del sistema operativo android, pero para efectos del presente proyecto se eligió como desarrollar la app para dispositivos que utilicen la desde la versión 4.0 hasta la 7.1.2, Esto con la finalidad de abarcar la mayor parte de dispositivos disponibles actualmente en el mercado.

8.4 Normalización de base de datos.

Debido a que la app almacenará datos, se optó por utilizar a SQLite como gestor de base de datos, ya que proporciona muchas ventajas para el desarrollo móvil.

La app CalculaTacos fue diseñada para ser utilizada en cualquier PYME encargada de la preparación y venta de alimentos, pero para efecto de ejemplificar su desarrollo y funcionalidad se partió de aplicar el desarrollo en una taquería. A continuación de describe de manera detallada los pasos a seguir en cuanto a la evolución del modelo de base de datos que se utilizó. Como en toda planeación de bases de datos se partió de una tabla (tabla 2) que contenía los datos y ejemplos del control de ventas.

Tabla 2. Tabla inicial requerimientos de la base de datos.

Mesero	Mesa	PedidoI	CantidadI	PedidoN	CantidadN	Total	Cliente
Carlos	Mesa1	Tacos	5	Refresco	1	57	cli-0023
Carlos	Mesa2	Refresco	1	Tacos	3	39	cli-0024
José	Mesa3	Tacos	1	Refresco	1	21	cli-0025
Pedro	Mesa4	Tacos	10	Refresco	1	102	cli-0026

Ejemplo del control de los datos que se manejan en una taquería.

Posteriormente se aplicó la normalización a la base de datos. En primera instancia se utilizó la primera forma normal. En este sentido, se intentó eliminar los datos repetidos tal y como se muestra en la Figura 5.

Producto		
Nombre	Precio	Cantidad
Tacos	9	10
Refrescos	12	1

Usuario	
idUsuario	Nombre
Me-01	Carlos
Me-02	José
Me-03	Pedro

Mesa				
Nombre	Usuario	Fecha	Productos	Cliente
Mesa1	Carlos	07/02/17	Tacos, Refrescos	cli-0023
Mesa2	Carlos	07/02/17	Tacos, Refrescos	cli-0024
Mesa3	José	07/02/17	Tacos, Refrescos	cli-0025
Mesa4	Pedro	07/02/17	Tacos, Refrescos	cli-0026

Cliente	
idNombre	Pedido
cli-0023	Mesa1
cli-0024	Mesa2
cli-0025	Mesa3
cli-0026	Mesa4

Figura 5. Normalización de la base de datos utilizando la 1FN.

Una vez aplicada la primera forma normal se procedió a utilizar la segunda forma de normalización en donde se modificaron las tablas para que se formara una base de datos que cumpliera con las características solicitadas. Se puede apreciar en la figura 6.

Producto		
idProducto	Nombre	Precio
Prod-01	Tacos	9
Prod-02	Refresco	12

Usuario	
idUsuario	Nombre
Usr-01	Carlos
Usr-02	José
Usr-03	Pedro

Mesa	
idMesa	Nombre
Mesa-01	Mesa1
Mesa-02	Mesa2
Mesa-03	Mesa3
Mesa-04	Mesa4

Cliente	
idCliente	Nombre
cli-0023	Alejandra
cli-0024	Grecia
cli-0025	Monse
cli-0026	Paola

Reservación				
idReservacion	idCliente	idMesa	idUsuario	Fecha
Reserv-001	cli-0023	Mesa-01	Usr-01	07/02/17
Reserv-002	cli-0024	Mesa-02	Usr-01	07/02/17
Reserv-003	cli-0025	Mesa-03	Usr-02	07/02/17
Reserv-004	cli-0026	Mesa-04	Usr-03	07/02/17

Venta			
idVenta	idProducto	Cantidad	idReservación
Venta-0001	Prod-01	5	Reserv-001
Venta-0002	Prod-02	1	Reserv-001
Venta-0003	Prod-01	3	Reserv-002
Venta-0004	Prod-02	1	Reserv-002

Figura 6. Normalización de la base de datos utilizando la 2FN.

Como se explicará más adelante, esta base de datos se diseñó hasta la cuarta iteración de la metodología scrum. A continuación, se describe el uso de dicha metodología

8.5 Iteración 1.

El C. Adelaido Espinoza Cortes, quien es el dueño de dicho establecimiento, solicitó el desarrollo de una aplicación móvil que lo ayudara a hacer los cobros de las cuentas de una manera más fácil y eficiente, puesto que los métodos tradicionales que utilizaba resultaban ser poco confiables.

En este sentido, en la tabla 3 se especifican los roles y el nombre de las personas que los ocuparan.

Tabla 3. Roles ocupados dentro de la metodología Scrum.

Rol	Persona
Stakeholder	C. Adelaido Espinoza Cortez.
Product Owner	José Francisco Vergara Espinoza.
Scrum Master	José Francisco Vergara Espinoza.
Team	José Francisco Vergara Espinoza.

Tabla que muestra los roles requeridos en la metodología Scrum ágil orientado al desarrollo de la app.

Debido a que el IDE de desarrollo “Android Studio” cuenta con muchas características que facilitan el trabajo de los desarrolladores, en el marco de la metodología scrum, un desarrollador puede tomar el papel de todos los puestos del equipo de desarrollo.

8.5.1 Planificación de la iteración 1.

En primera instancia el C. Adelaido Espinoza Cortez, en calidad de cliente ó stakeholder, en una reunión casual de no más de una hora, solicitó una aplicación que le ayudara a realizar el cobro de pedidos en su taquería de una forma certera y al mismo tiempo que se pudiera instalar en su teléfono celular de gama media, el cual hacía uso del sistema operativo android. Los requerimientos que él solicitaba en aquella ocasión eran los siguientes:

8.5.1.1 Selección de requisitos.

- Una aplicación a la medida que calculara el costo total de los productos que él vendía. En este caso el número de productos que el señor comercializaba en ese momento era de tres.
- La aplicación solo debía calcular el total. No tomando en cuenta los subtotales de cada uno de los productos vendidos.

- La aplicación no debería tener ningún error al generar el total.
- El usuario de la aplicación debería ingresar por el teclado la cantidad de los productos vendidos.
- Que la interfaz gráfica sea fácil de entender y utilizar.
- Que tuviera un bajo costo.
- Que el tiempo promedio de respuesta fuera menor en comparación con los métodos tradicionales que él utilizaba.

Por otra parte, después de obtener los requisitos del cliente se procedió a realizar una encuesta que constaba de once preguntas a un total de cien PyMES distribuidoras de comida rápida. Con el fin de detectar un mayor número de requisitos que compartieran este tipo de negocios, para finalmente poder mejorar la app y que de esta manera tuviera un mayor grado de estandarización y para poder ser utilizada por la mayor parte de los negocios encargados de la preparación y venta de alimentos y bebidas.

En la encuesta anteriormente mencionada, se obtuvieron resultados interesantes que efectivamente demuestran que existen errores al generar la cuenta de un cliente en horarios pico, ya que el 89% de las empresas encuestadas afirmaba que los errores eran recurrentes cuando la clientela es fluida. Por otra parte, los métodos más recurrentes eran: utilizando lápiz y papel, con calculadora estándar y haciendo las cuentas mentalmente; ningún negocio afirmó hacer uso de algún sistema informático para llevar el control de alguno de sus procesos.

La encuesta también reveló algo interesante en cuanto a la interfaz de comunicación con algún sistema que les ayudara a automatizar su proceso de venta, dado que el 73% de los negocios encuestados manifestaba que preferían interactuar con el sistema por medio de comandos de voz en lugar de utilizar el teclado.

8.5.1.2. Gestión del backlog.

Posteriormente de que el cliente expresara los requerimientos, se procedió a generar las historias de usuario junto con la estimación de cada una de ellas en el Sprint Backlog. En la

tabla 4 se establecen las historias de usuario tratando de representar cada uno de los requisitos solicitados.

Tabla 4. Historias de usuario (Iteración 1) .

	Historias de usuario	Estimación
1	Como usuario, quiero que la app tenga registrados los nombres de cada uno de los productos que vendo.	2
2	Como usuario, quiero que la app tenga registrados los precios de cada uno de los productos que comercializo.	2
3	Como usuario, quiero que la app tenga en frente del nombre de cada producto un campo de “cantidad” en donde por medio del teclado pueda ingresar la cantidad de productos vendidos en cada uno de mis pedidos.	3
4	Como usuario, quiero que la app tenga un botón llamado “Calcular”, el cual al presionarlo multiplique el campo “Cantidad” por el precio de cada uno de mis productos.	3
5	Como usuario, quiero que la app tenga un campo en donde después de ser presionado el botón “Calcular” aparezca en números la cantidad total a pagar por la cuenta generada.	2
6	Como usuario, quiero que la app sea fácil de utilizar.	3

7	Como usuario, quiero que la app sea entendible en cuanto al tipo de letra y colores.	3
8	Como usuario, quiero que la app calcule el total a cobrar en cada uno de mis pedidos, para reducir el tiempo de respuesta y no errar al hacer las operaciones aritméticas.	18

Historias de usuario de la iteración 1, en donde se muestra que 18 es la estimación de complejidad de las tareas establecidas.

8.5.2 Ejecución de la iteración 1.

Posteriormente de la planificación del sprint se procedió a desarrollar la primera aplicación tratando de satisfacer todos los requerimientos planteados al inicio. Inicialmente se empezó a desarrollar un proyecto en el IDE de desarrollo Android Studio. La primera aplicación era bastante sencilla, en la parte de la interfaz gráfica solo contaba con tres campos para rellenar:

1. Tacos normales.
2. Tacos de lengua.
3. Refrescos.

Cabe mencionar que dichos campos se tomaron en cuenta de acuerdo a los diferentes productos y precios que comercializaba el stakeholder. También se podía observar un texto “Total:” y un botón llamado “Calcular”. Este último activaba toda la programación en java la cual era relativamente simple. Constaba de dos operaciones, la primera se encargaba de obtener los subtotales de cada producto, esto quiere decir que en cada producto se multiplicaba la cantidad ingresada en los campos por el costo de cada producto. En cuanto a la segunda operación, era simplemente sumar los subtotales obtenidos anteriormente y desplegar el total en la parte de abajo en números de color rojo. En la figura 7 se puede apreciar la aplicación ejecutada en un dispositivo android, mientras que en la figura 8 se muestra la misma aplicación, pero con unos datos de prueba para generar un ejemplo de su uso real.



Figura 7 App "CalculaTacos" ejecutada (iteración 1).



Figura 8 App "CalculaTacos" con datos "iteración 1".

Como se mencionaba anteriormente, esta aplicación era bastante simple, debido a que carecía de todas las validaciones necesarias para evitar que ésta se cerrara inesperadamente al momento de su ejecución. Sin embargo, esta aplicación estaba lista para mostrársela al stakeholder que estaba muy ansioso de probarla en su negocio.

8.5.3 Entrega de la primera iteración.

Después de dos semanas de desarrollo, se contactó al stakeholder para mostrarle el primer incremento de su aplicación móvil. Ese mismo día se le hicieron saber de manera general las características y funcionalidades de las cuales su aplicación tenía alcance hasta ese momento. Después, se le instaló la aplicación en su teléfono personal, el cual era un dispositivo android de gama media, que contaba con android lollipop 5.0. A partir de ese día el stakeholder utilizó la aplicación en su negocio durante una semana. Este tiempo sirvió para que la app estuviera a prueba en un entorno de uso real y así se pudieran detectar fallas o posibles mejoras en su funcionamiento.

8.5.4 Feedback.

Transcurrió una semana completa y se tuvo una tercera reunión con el stakeholder con la finalidad de conocer su punto de vista acerca de su aplicación. El stakeholder inició la conversación con un notable semblante de satisfacción, comentó que la aplicación le había sido realmente útil en la semana que la tuvo a prueba en su negocio, porque ésta realizaba el cálculo de las cuentas mucho más rápido y en ninguna ocasión arrojó algún resultado erróneo. Prácticamente, hasta ese momento el stakeholder estaba convencido de lo beneficioso que puede llegar a ser el adoptar la tecnología en procesos muy repetitivos.

Sin embargo, una vez transcurridos unos minutos conversando acerca de la primera aplicación, el stakeholder hizo tres comentarios muy importantes, los cuales fueron pilares para el desarrollo del proyecto final. El primer comentario fue acerca de la posibilidad de poder agregar más productos a la aplicación, debido a que, dependiendo de la temporada del año, el stakeholder tiende a comercializar más ó menos productos. En cuanto al segundo comentario fue acerca de poder cambiar la forma en que el usuario se comunica con la

aplicación, debido a que, para el stakeholder le parecía muy pequeño el teclado de su teléfono celular. Lo cual dificultaba la utilización de la aplicación móvil. Por último, el tercer comentario fue acerca de poder visualizar los subtotales y el total posteriormente de presionar el botón calcular, para conocer el monto de cada tipo producto, a diferencia de la primera aplicación en donde sólo se mostraba el total.

8.6 Iteración 2.

8.6.1 Planificación de la iteración 2.

Como se mencionaba anteriormente, el stakeholder contribuyó a la retroalimentación en la primera iteración de la aplicación móvil. En dicha retroalimentación fue posible replantear los requerimientos para el desarrollo de la aplicación móvil, debido a que hasta ese momento se necesitaban satisfacer nuevas necesidades del stakeholder. En este sentido, a continuación, se describen de forma general los requerimientos a tomar en cuenta para la segunda iteración.

8.6.1.1 Selección de requisitos.

- Una aplicación a la medida que calculara el costo total de los productos que él vendía por medio de comandos de voz.
- La aplicación debería tener una base de datos para almacenar n número de productos.
- Que la aplicación tuviera las opciones de agregar, eliminar, actualizar y mostrar los productos.
- La aplicación debería mostrar en la pantalla los subtotales y el total a pagar.
- La aplicación no debería tener ningún error al generar los subtotales ni el total en términos monetarios.
- Que la interfaz gráfica sea fácil de entender y utilizar.
- Que tuviera un bajo costo.

- Que el tiempo promedio de respuesta fuera menor en comparación con los métodos tradicionales que él utilizaba.

8.6.1.2 Gestión del backlog.

De acuerdo a los requisitos recabados en la reunión de feedback con el stakeholder, se procedió a generar las historias de usuario. En la tabla 5 se muestran cada una de las historias de usuario junto a su estimación.

Tabla 5. Historias de usuario (Iteración 2).

	Historias de usuario	Estimación
1	Como usuario, quiero que la app tenga una base de datos para agregar todos los productos que manejo.	8
2	Como usuario, quiero que la app tenga una opción para agregar productos a la base de datos.	13
3	Como usuario, quiero que la app tenga una opción para eliminar los productos de la base de datos.	13
4	Como usuario, quiero que la app tenga una opción de actualizar los atributos de cada producto agregado previamente.	13
5	Como usuario, quiero que la app tenga una opción para mostrar todos los productos que se encuentran registrados en la base de datos.	8
6	Como usuario, quiero que la app pueda almacenar por lo menos 100 productos.	5
7	Como usuario, quiero que la app funcione con comandos de voz simples.	40

8	Como usuario, quiero que la app calcule y arroje en la pantalla los subtotales de cada uno de los productos vendidos.	8
9	Como usuario, quiero que la app calcule y arroje el total de cada venta realizada.	8
10	Como usuario, quiero que la app escriba en la pantalla las palabras que ingreso por medio del micrófono, para saber si el teléfono ha escuchado correctamente.	8
11	Como usuario, quiero que la app sea fácil de utilizar.	8
12	Como usuario, quiero que la app sea entendible en cuanto al tipo de letra y colores.	13
13	Como usuario, quiero que la app calcule subtotales y el total a cobrar en cada uno de mis pedidos, por medio de comandos de voz simples, para reducir el tiempo de respuesta y no errar al hacer las operaciones aritméticas de la forma tradicional.	145

Historias de usuario de la iteración 2, en donde se muestra que 145 es la estimación de complejidad de las tareas establecidas.

8.6.2 Ejecución de la iteración 2.

Después de haber planteado en el backlog cada una de las historias de usuario seguidas de su estimación, se procedió a desarrollar la aplicación en su segunda versión, debido a que los nuevos requerimientos fueron lo suficientemente disruptivos para plantear de una nueva manera la parte gráfica y lógica de la aplicación.

Al igual que en la primera iteración de este desarrollo, la aplicación se desarrolló en el IDE android studio. Sin embargo, en esta ocasión fue necesario implementar una nueva clase llamada SQLiteOpenHelper, debido a que es necesaria para poder crear y gestionar la base de datos que utilizará la aplicación para almacenar los productos que comercialice el

stakeholder. Es importante señalar que dicha aplicación utilizara a SQLite como motor de base de datos.

Como paso siguiente, se procedió a implementar los métodos necesarios para poder crear la base de datos con ayuda de la clase SQLiteOpenHelper. Una vez generada la base de datos fue necesario comenzar a diseñar la interfaz gráfica tomando en cuenta que una de sus características debía ser la facilidad de uso, para que cualquier persona mayor a 8 años la pudiera utilizar sin ningún problema. En este sentido, la interfaz gráfica estaba formada por cinco pantallas ó actividades diferentes.

1. La primer Activity era la principal (figura 9), debido a que en esta era donde se iba a llevar a cabo el dictado de comandos, mostrar lo escuchado por el usuario, mostrar el subtotal y total de la cuenta generada. Además de eso, la aplicación contaba por defecto con un botón en forma de micrófono que activaba el reconocimiento de voz y un botón llamado Productos, el cual direccionaba a la Activity 2.



Figura 9 App "CalculaTacos" ejecutada y con datos (iteración 2).

2. La Activity 2 llamada Productos (figura 10) era la encargada de mostrar tres diferentes métodos para administrar la base de datos de la aplicación. Estos métodos tenían la función de visualizar, agregar y eliminar productos de la base de datos

respectivamente. Cada uno de ellos direccionaba a otra Activity que ejecutaba los métodos propios de cada función.



Figura 10 App "CalculaTacos" Activity Productos (iteración 2).

3. La Activity 3 (figura 11) era la encargada de mostrar todos los productos de la base de datos. Esta función era tan simple, pero a la vez útil que posteriormente se incluyó en otras activities.

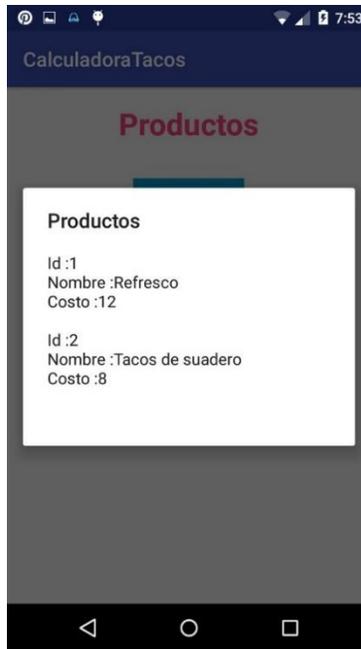


Figura 11 App "CalculaTacos" Activity Productos-Ver Productos (iteración 2).

4. La Activity 4 (figura 12) era la encargada de agregar productos a la base de datos. Estaba conformada por cinco elementos; el título, el campo nombre, el campo costo, el botón ver productos y el botón agregar. Para poder agregar un nuevo producto, el usuario tenía que ingresar los dos campos solicitados y presionar el botón agregar.



Figura 12 App "CalculaTacos" Activity AgregarProducto (iteración2).

5. La Activity 5 (figura 13) era la encargada de eliminar algún producto. Ésta constaba de 5 elementos; el título, un texto con las instrucciones para eliminar algún producto, un campo para escribir el id del producto a eliminar, el botón de ver id's y por último el botón de eliminar. El procedimiento para poder eliminar algún producto era relativamente sencillo, simplemente era necesario ingresar el campo id... el id del producto a eliminar y finalmente presionar el botón eliminar para completar el proceso.

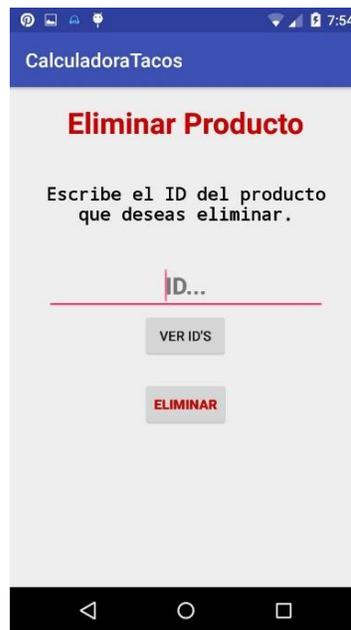


Figura 13 App "CalculaTacos" Activity EliminarProducto (iteración2).

Como se mostró anteriormente, se intentó desarrollar la interfaz gráfica para que fuera muy intuitiva y fácil de usar. El siguiente paso fue programar la parte lógica de la aplicación. De este modo, es importante mencionar que se generó una clase java para cada Activity debido a que cada una tenía que ejecutar por lo menos algún método. En cuanto a los métodos de visualizar, agregar y eliminar productos se generaron las consultas en lenguaje SQL dentro de la clase SQLiteOpenHelper, las cuales realizaban las tareas necesarias para cumplir con las funciones propias de cada método. Por otro lado, en la parte lógica de reconocimiento de voz se utilizó la clase "public class SpeechRecognizer". Esta clase proporciona el acceso al

reconocimiento de voz. La función de reconocimiento de voz se activaba en la clase MainActivity.java, específicamente al presionar el botón en forma de micrófono.

A continuación, se describe un diagrama (figura 14) del algoritmo que se utilizó para reconocer los comandos de voz.

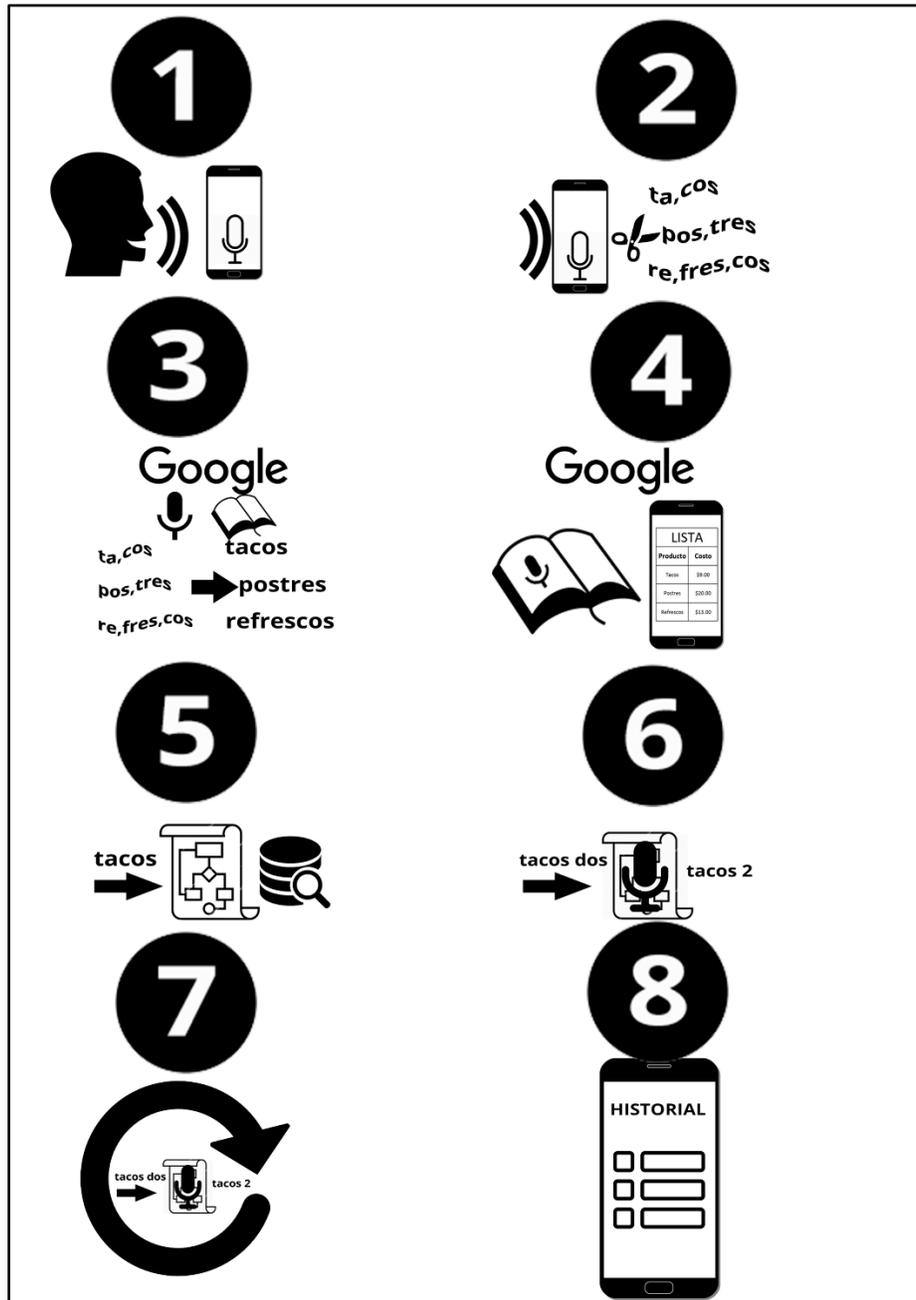


Figura 14. Algoritmo utilizado para reconocer las instrucciones de voz que el usuario envía. Fuente propia.

1. La clase “public class SpeechRecognizer” proporciona un string con el texto escuchado a presionar el botón en forma de micrófono p. ej. “Tacos 10 Refrescos 1 Aguas 2”.
2. Se crea un array llamado “palabras” de N elementos, el cual se rellena con palabras individuales obtenidas del string inicial. Tomando en cuenta el ejemplo anterior, el arreglo palabras quedaría como se muestra en la tabla 8.

Tabla 8. Array "palabras".

Tacos	10	Refrescos	1	Aguas	2
0	1	2	3	4	5

Ejemplo de el array generado por un pedido de diferentes productos.

3. Se creaba un ciclo for para recorrer el arreglo y buscar los nombres de los productos y la cantidad vendida.
4. Cada vez que se posicionaba en un elemento en un producto se procedía a buscar dentro de la base de datos alguna coincidencia, si esta aparecía, se obtenía el costo en la base de datos y se multiplicaba por la cantidad.
5. Finalmente se sumaban todos los subtotales para generar el total general.
6. Como último paso se procedía a mostrar en un mensaje los subtotales y el total.

8.6.3 Entrega de la iteración 2.

Una vez completado el segundo desarrollo de la aplicación solicitada, se tuvo una reunión con el stakeholder en donde se le mostró la aplicación. Además de instalarle la aplicación se le dio una breve capacitación para utilizar su aplicación.

Al igual que en la primera entrega, el stakeholder se comprometió a probar la aplicación durante una semana en su negocio. Este tiempo serviría para poder adaptarse a su nueva aplicación con reconocimiento de voz y poder detectar posibles fallos no contemplados en el desarrollo.

8.6.4 Feedback.

Como se había previsto, una semana más tarde se sostuvo una reunión con el stakeholder teniendo como finalidad la retroalimentación en torno al desarrollo de la aplicación móvil.

El stakeholder comenzó hablando muy entusiasmado acerca de lo fácil que era utilizar la aplicación con reconocimiento de voz. Comentaba que no tenía dificultad al momento de visualizar, agregar ó eliminar productos de la base de datos. En ese aspecto la aplicación era bastante intuitiva. Por otra parte, comenzó a hablar acerca de que en determinadas ocasiones la aplicación se cerraba inesperadamente arrojando el mensaje “Lo sentimos, la aplicación se detuvo”. Después de cuestionarlo acerca de las ocasiones específicas en donde la aplicación tendía a cerrarse inesperadamente, fue posible notar que se debía a la falta de validaciones en cuanto al texto escuchado y por consiguiente el error se manifestaba en la parte lógica de las operaciones aritméticas. Otro aspecto que mencionó el stakeholder fue acerca de los colores utilizados en la interfaz gráfica, si bien es cierto que los colores eran llamativos, el cliente sugirió la posibilidad de hacer ver más formal a la aplicación. Esto era simplemente estética.

Pero en general el cliente estaba muy satisfecho en cuanto al desarrollo realizado y específicamente el cuanto a la utilización de comandos de voz para poder calcular las cuentas en un menor tiempo del que se tardaba anteriormente. En otras palabras, el feedback fue bastante favorable, puesto que los cambios a realizar eran mínimos hasta ese momento.

8.7 Iteración 3.

8.7.1 Planificación de la iteración 3.

De acuerdo a la retroalimentación generada durante la última reunión con el stakeholder fue posible recabar los requisitos importantes para proceder a desarrollar el incremento de la aplicación móvil, que, si bien es cierto, en el feedback anterior no hubo cambios estructurales, fue posible encontrar algunos detalles de validación de datos y otros concernientes a la

interfaz gráfica. A continuación, se describen los nuevos requisitos para optimizar la aplicación.

8.7.1.1 Selección de requisitos.

- Validar si el usuario ingresa por lo menos alguna palabra por medio del reconocimiento de voz.
- Validar si la cantidad del producto es un número.
- Validar si la sintaxis está completa.
- Validar si el producto detectado en el texto se encuentra en la base de datos.
- Crear una nueva clase llamada producto para poder manejar los datos de producto de una manera más fácil y reducir el tiempo de respuesta.
- Cambiar el color de los botones a uno más formal.
- Cambiar la fuente utilizada en toda la aplicación por una más formal.
- Diseñar e implementar un logotipo.

8.7.1.2 Gestión del backlog.

De acuerdo a los requisitos recabados en la reunión de feedback con el stakeholder, se procedió a generar las historias de usuario. En la tabla 9 se muestran cada una de las historias de usuario junto a su estimación.

Tabla 9. Historias de usuario (Iteración 3).

	Historias de usuario	Estimación
1	Como usuario, quiero que la app valide si escucha por lo menos una palabra.	3
2	Como usuario, quiero que la app detecte si en cantidad ingresé un número y no letras.	3
3	Como usuario, quiero que la app me recuerde la sintaxis a utilizar si es que yo utilizo una equivocada.	3

4	Como usuario quiero que la aplicación valide si el producto que ingresé no se encuentra en la base de datos.	3
5	Como usuario, quiero que la app realice el cálculo de la cuenta en el menor tiempo posible.	13
6	Como usuario, quiero que los botones tengan un aspecto más formal.	3
7	Como usuario, quiero que la fuente utilizada en la app tenga un aspecto más formal.	3
8	Como usuario, quiero que la aplicación tenga un logotipo.	8
9	Como usuario, quiero que la app calcule subtotaes y el total a cobrar en cada uno de mis pedidos, por medio de comandos de voz simples, para reducir el tiempo de respuesta y no errar al hacer las operaciones aritméticas de la forma tradicional.	39

Historias de usuario de la iteración 3, en donde se muestra que 39 es la estimación de complejidad de las tareas establecidas.

8.7.2 Ejecución de la iteración 3.

Para darle continuidad al desarrollo de la aplicación, se procedió a cambiar la interfaz gráfica, en otras palabras, se intentó darle un aspecto más formal a cada una de las actividades tomando como referencia la psicología del color.

La estructura de la aplicación quedó intacta, solo se realizaron algunos cambios en cuanto al tipo de letra y a los colores utilizados.

1. La primer Activity (figura 15) conservó su nivel jerárquico, ya que en esta se realiza el dictado de comandos, mostrar lo escuchado por el usuario, mostrar el subtotal y total de la cuenta generada. Por otra parte, se eliminó el nombre de la aplicación de la parte superior, para aumentar el espacio de escritura de los comandos de voz ingresados.

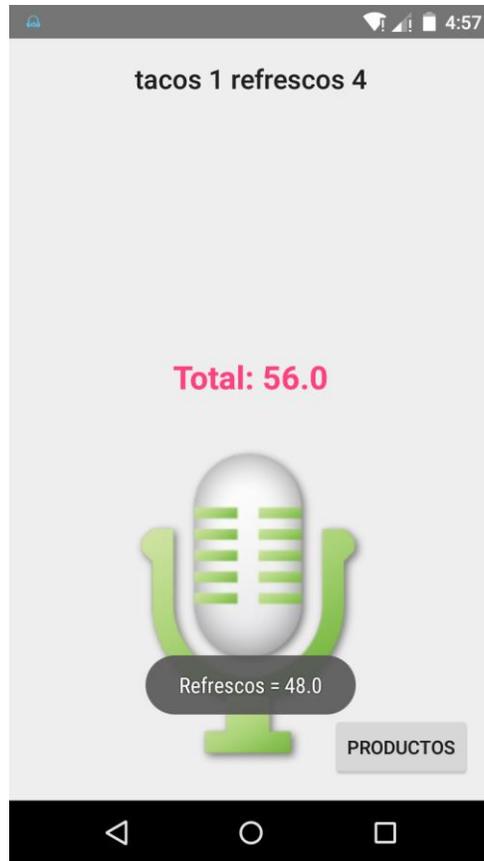


Figura 15 App "CalculaTacos" ejecutada y con datos (iteración 3).

2. La Activity 2 (figura 16) llamada Productos es la encargada de mostrar tres diferentes métodos para gestionar la base de datos de la aplicación. Estos métodos tenían la función de visualizar, agregar y eliminar productos de la base de datos respectivamente. En esta Activity solo se le aplicó un estilo formal al diseño de los botones y el tipo de letra.



Figura 16 App "CalculaTacos" Activity Productos (iteración 3).

3. La Activity 3 (figura 17) es la encargada de mostrar todos los productos de la base de datos. Esta función era tan simple, pero a la vez útil que posteriormente se incluyó en otras activities. Prácticamente en esta Activity no se le aplicó ningún cambio respecto a la versión anterior.

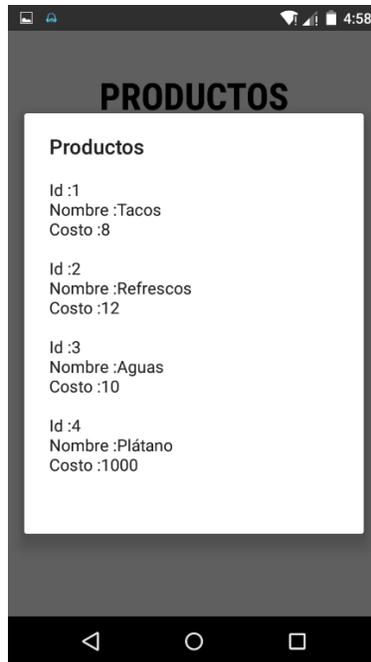


Figura 17 App "CalculaTacos" Activity Productos-Ver Productos (iteración 3).

4. La Activity 4 (figura 18) es la encargada de agregar productos a la base de datos. Estaba conformada por cinco elementos; el título, el campo nombre, el campo costo, el botón ver productos y el botón agregar. Para poder agregar un nuevo producto, el usuario tiene que rellenar los dos campos solicitados y presionar el botón agregar. Continuando con la aplicación de estilos, se le dio un diseño formal a los botones, color y tipo de letra.

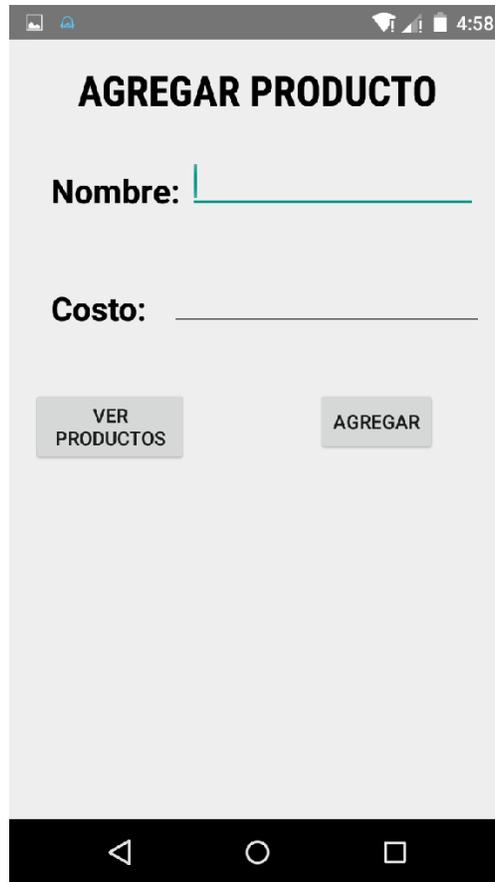


Figura 18 App "CalculaTacos" Activity AgregarProducto (iteración 3).

5. La Activity 5 (figura 19) es la encargada de eliminar algún producto. Ésta consta de 5 elementos; el título, un texto con las instrucciones para eliminar algún producto, un campo para escribir el id del producto a eliminar, el botón de ver id's y por último el botón de eliminar. El procedimiento para poder eliminar algún producto es sencillo, simplemente se necesita ingresar el campo "id... ", el id del producto a eliminar y finalmente presionar el botón eliminar para completar el proceso.

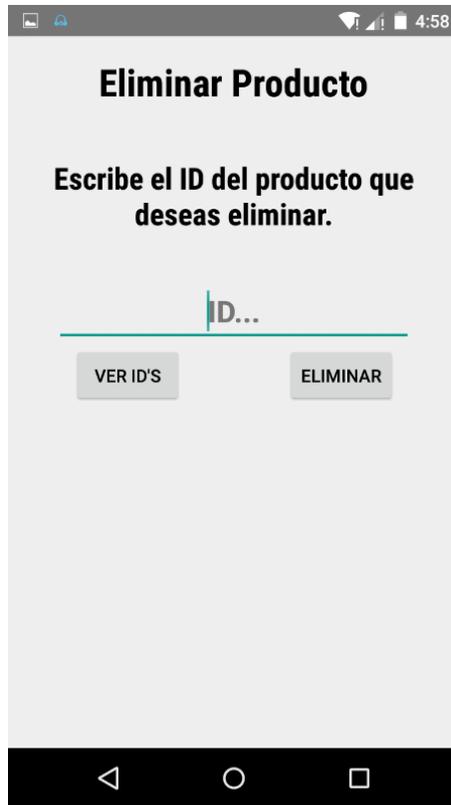


Figura 19 App "CalculaTacos" Activity EliminarProducto (iteración 3).

6. Es necesario comentar que la opción de actualizar producto está incluida en la Activity "Agregar producto" (figura 20). Esta función se activa cuando se intenta agregar un producto que ya existe en la base de datos.

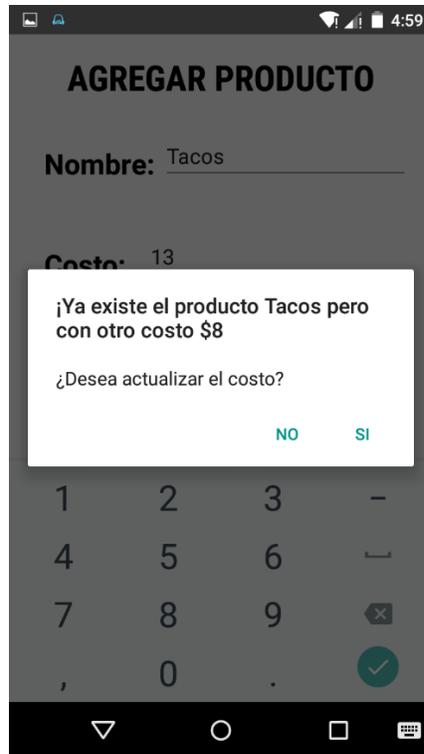


Figura 20 App "CalculaTacos" Actualizar Producto (iteración 3).

En cuanto a la validación de la sintaxis se tomaron en cuenta tres tipos diferentes de estas:

- Si no se escuchó ninguna palabra.
- Si la cantidad del producto contiene letras.
- Si la sintaxis es errónea.

Con respecto a la modificación del algoritmo, se generó una nueva clase llamada “Producto” la cual facilita el uso en los datos manejados a través de la base de datos. Así mismo agiliza el cálculo de las operaciones aritméticas y la representación en la pantalla de los subtotales y el total general de cada una de las cuentas.

Por último, se realizó el diseño del logotipo que iba a identificar a la aplicación en la pantalla del dispositivo en el que estuviera instalada. Para el diseño se tomaron en cuenta diferentes factores como:

- Color.
- Tipo de imagen.

- Imágenes.
- Simplicidad

Es por ello que se realizó un diseño discreto, que identificara fácilmente la función que realizaba la aplicación. Éste es un logotipo vectorial en el que se aprecian básicamente tres elementos gráficos: el primero es un cuadrado con esquinas redondeadas en color azul, este cuadro se toma como fondo. En segundo lugar, se aprecia una cabeza humana con unas líneas a la altura de la boca que representan el sonido emitido por la boca. En tercer lugar, aparece una calculadora en la cabeza humana antes mencionada, haciendo referencia las operaciones matemáticas que la aplicación ejecuta. En la figura 21 se muestra el logotipo utilizado para identificar la aplicación “CalculaTacos”.



Figura 21 Logotipo de app "CalculaTacos".

Como ultima característica anexada al presente desarrollo, se le dio el nombre de “CalculaTacos” a la aplicación móvil, este nombre fue elegido tomando en cuenta la funciones que esta cumplía y el segmento de mercado a quien estaba dirigida en primera instancia, además de que el tamaño del nombre fuera corto para que los usuarios la identificaran más rápido.

8.7.3 Entrega de la iteración 3.

Posterior al término de la ejecución de la tercera iteración, habiendo tomado en cuenta todos los requerimientos planteados por el stakeholder y algunas modificaciones implementadas por el equipo de desarrollo; la aplicación estaba lista para ser entregada al stakeholder. Así que se tuvo una reunión para hacer la entrega del incremento al cliente y hablar acerca de los pormenores suscitados durante la iteración número tres.

De manera general, se le comentó al stakeholder que se había corregido la parte en donde se cerraba la aplicación repentinamente debido a la falta de validaciones de datos. Por otra parte, se le mostro la nueva interfaz gráfica que la app tenía. Y se le hicieron algunas pruebas del uso de la aplicación para que el stakeholder pudiera notar los cambios que esta había tenido.

Al finalizar la reunión, se acordó que el stakeholder iba a probar la aplicación durante una semana en su negocio al igual que en las dos iteraciones anteriores. Y finalmente se agendo otra reunión al finalizar el periodo de prueba.

8.7.4 Feedback.

Una semana más tarde se llevó a cabo la reunión agendada previamente para que tuviera lugar la retroalimentación de la tercera iteración. Esta reunión fue muy grata, debido a que el stakeholder estaba muy entusiasmado con su nueva adquisición tecnológica en su negocio. Comentaba que la aplicación satisfacía todas las necesidades en cuanto a calcular cuentas se refiere, porque durante la semana de pruebas no se había cerrado en ninguna ocasión a aplicación. Por otra parte, el stakeholder expresó que el nuevo diseño de la interfaz gráfica le había agradado bastante, incluso el logotipo y el nombre le parecía muy serio considerando que representaba de buena forma las funciones que llevaba a cabo la aplicación móvil.

En otras palabras, la aplicación había sido aceptada totalmente por el stakeholder porque cumplía con todos los requerimientos planteados por el mismo. Al terminar esta reunión se dio por concluido el desarrollo de la app “CalculaTacos” para los requisitos que el stakeholder había solicitado.

8.8 Iteración 4.

8.8.1 Planificación de la iteración 4.

Posteriormente de que la app había sido aceptada por el stakeholder de manera satisfactoria se procedió a hacer unas mejoras sustanciales a la aplicación. Se tomaron en cuenta factores recabados en la encuesta realizada, además de incluir elementos como mesas y el usuario de la app. Todo esto con el fin de hacer un sistema orientado a satisfacer las necesidades de la mayoría de los negocios encargados de la preparación y venta de alimentos. A continuación, se describen los nuevos requisitos para optimizar la aplicación.

8.8.1.1 Selección de requisitos.

- Agregar la opción de mesas.
- Tener la opción de pagar cada una de las mesas que se encuentren activas.
- Crear un usuario para cada app.
- Generar un historial de las ventas realizadas de acuerdo a fechas personalizadas.
- Escalar la base de datos con la finalidad de que la app pueda generar reportes de las ventas históricas.

8.8.1.2 Gestión del backlog.

De acuerdo a los requisitos anexados por parte de las encuestas, se procedió a generar las historias de usuario. En la tabla 10 se muestran cada una de las historias de usuario junto a su estimación.

Tabla 10. Historias de usuario (Iteración 4).

	Historias de usuario	Estimación
1	Como usuario, quiero que la app pueda llevar el control de varias mesas al mismo tiempo.	10

2	Como usuario quiero que la app tenga un botón de “pagar” cada una de las mesas que se encuentren activas.	15
3	Como usuario, quiero que la app tenga su propio usuario.	10
4	Como usuario, quiero que la app genere un historial de las ventas realizadas.	15
5	Como usuario, quiero que la app tenga la capacidad de poder llevar el control de diferentes mesas al mismo tiempo, además de poder obtener un reporte con las ventas realizadas.	50

Historias de usuario de la iteración 4, en donde se muestra que 50 es la estimación de complejidad de las tareas establecidas.

8.8.2 Ejecución de la iteración 4.

En la figura 22 se puede observar la Activity principal, en donde se procedió a agregar un elemento spinner el cual permite seleccionar las diferentes mesas disponibles. Además, se agregaron dos nuevos botones, uno correspondiente a “Reporte” y el otro a “Pagar”.

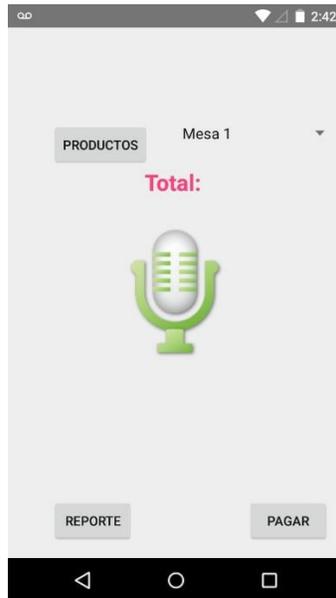


Figura 22. Activity Principal con elemento Spinner.

En la Activity de reporte que se muestra en la figura 23, está diseñada para poder mostrar las ventas históricas realizadas. Una particularidad interesante es que dicha Activity tiene la posibilidad de poder filtrar las ventas históricas del periodo de tiempo que se necesite. Además del elemento spinner dedicado a seleccionar la mesa de la cual se necesite el reporte.

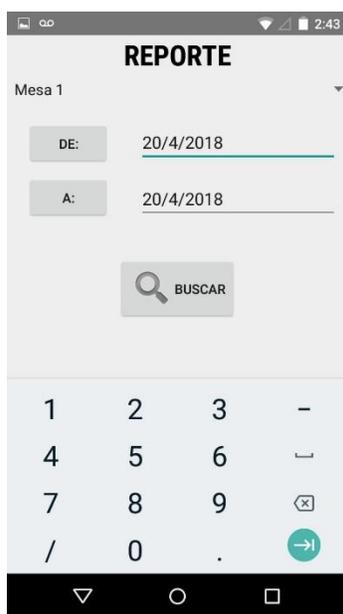


Figura 23. Activity Reporte, Iteración 4.

La figura 24 muestra la última Activity que se agregó fue la Activity Pagar, Dicha Activity está conformada nuevamente por un spinner de mesas, un campo total, un campo pago y un campo cambio; el usuario de la app solo tendrá que seleccionar la mesa, e ingresar el monto con el que el cliente paga y por último registrar la venta con el botón pagar.

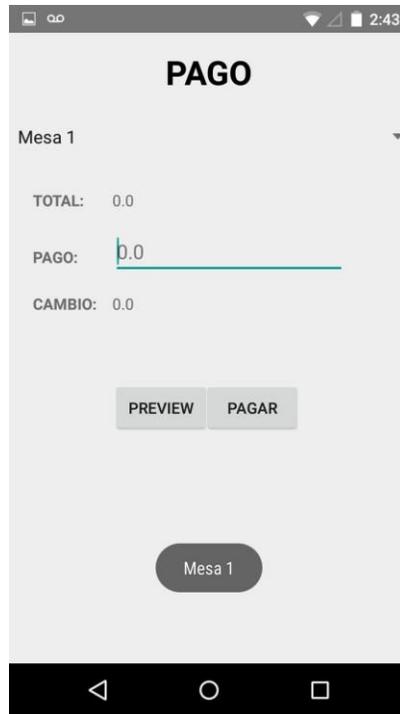


Figura 24. Activity Pago, Iteración 4.

Después de generar la interfaz gráfica se generó la parte de programación correspondiente para cada una de las activities correspondientes, además de emplear el modelo MVC para gestionar el proyecto en android.

8.8.3 Entrega de la iteración 3.

Posterior al término de la ejecución de la cuarta iteración, habiendo tomado en cuenta todos los requerimientos de mejora se llevaron a cabo diferentes pruebas de funcionalidad, en donde la app salió adelante dado que la app ejecuta las operaciones aritméticas en milésimas de segundo, además de que los reportes que ésta brinda son muy útiles para los administradores de las PYMES encargadas de la preparación y venta de alimentos para conocer el monto total de las ventas correspondientes a un periodo de tiempo determinado.

8.8.4 Feedback.

La retroalimentación correspondiente a esta última iteración se llevó a cabo contactando nuevamente al stakeholder para que probara la versión mejorada de la app que inicialmente se había desarrollado para él.

Después de haber probado la app en su negocio durante una semana, se realizó una reunión con él para comentar la funcionalidad de la app. Realmente los comentarios fueron satisfactorios ya que el stakeholder manifestó que la app seguía siendo igual de práctica que su versión anterior, además de que en esta ocasión la app le permitía conocer las ventas realizadas en su negocio durante un periodo de tiempo determinado.

8.9 Publicación de la app en Play Store.

Posterior al desarrollo de la aplicación “CalculaTacos”, se procedió a publicar dicha app en un medio electrónico para su difusión. En este caso se publicó en la tienda oficial de aplicaciones creada por Google. Se optó por este medio debido a las ventajas que tiene en cuanto a la difusión de apps.

Con el paso del tiempo Play Store ha adquirido gran prestigio entre los usuarios de smartphones gracias a las características que tiene la tienda. En otras palabras, Play Store es un medio confiable de difusión de contenidos (Aplicaciones, libros, juegos, etcétera), dado que los altos estándares de Google exigen que las aplicaciones sean de calidad y no supongan ningún riesgo para los usuarios.

Una vez publicada la app generada en el presente proyecto “CalculaTacos”, se puede concluir este trabajo de investigación de manera satisfactoria. Ya que se cumplieron los objetivos planeados inicialmente. Éstos empezaron con la intención de apoyar a un comerciante de alimentos y bebidas, y culminó con la publicación de la app en la tienda Play Store, la cual debido a su alcance mundial hace posible que la app se encuentre disponible en prácticamente todo el mundo en dónde un dispositivo android tenga conexión a internet.

8.10 Pruebas

Una vez desarrollada la app CalculaTacos y probada en la taquería mencionada, se procedió a ejecutar tres tipos de pruebas. La primera prueba es acerca del tiempo que tarda un mesero en calcular la cuenta mediante los diferentes métodos tradicionales contra la app. La segunda prueba trata de evaluar el costo en términos monetarios que se generan al cobrar mal una cuenta. Por último, está la prueba de satisfacción del usuario, en donde se realizó una encuesta a diferentes usuarios después de haberla utilizado durante 7 días. A continuación, se describen los experimentos realizados:

8.10.1 Pruebas de tiempo.

La prueba de tiempo se realizó en la taquería Espartacos ubicada en el municipio de Tepetlaoxtoc en donde se le pidió al mesero que llevara a cabo el cálculo de 400 cuentas haciendo uso de los siguientes métodos (100 cuentas por cada método).

- App CalculaTacos.
- Lápiz y papel.
- Calculadora Estándar.
- Mentalmente

Para comprender mejor como está distribuida la taquería en donde se llevaron a cabo las pruebas a continuación se describen los elementos por los que estaba constituida:

- 1 mesero.
- 1 cocinero.
- 4 mesas.
- 16 bancos.
- 2 productos con diferentes precios.

8.10.2 Prueba de costos.

Tomando como referencia las 100 pruebas por cada uno de los métodos mencionados, hubo un observador que realizaba las cuentas a la par del mesero, y registro cada uno de los errores

de cálculo cometidos en cada método. En este sentido, se realizó una comparación de la efectividad de cada uno de los métodos, y así mismo estos resultados se tomaron como base para pronosticar las pérdidas en términos monetarios que se generan a cobrar mal una cuenta.

8.10.3 Satisfacción del usuario.

Como tercer parámetro para medir los resultados de la app, se consideró realizar una encuesta de satisfacción del usuario final. En este caso, la app CalculaTacos fue instalada en el celular de 20 meseros de los negocios que en primera instancia se obtuvieron las entrevistas. Con el fin de que utilizaran la app durante una semana y al terminar ese periodo se les aplicó una encuesta para conocer su punto de vista acerca de la app.

IX RESULTADOS

Para concluir el presente proyecto, es trascendente, expresar los resultados obtenidos después de realizar todo el trabajo de desarrollo e investigación. En este sentido se pretende mostrar cuatro parámetros para dar a conocer los efectos que se obtuvieron después de utilizar una metodología de desarrollo para aplicaciones móviles y hacer uso de dicha app para solventar un problema tan común en el ámbito de los negocios encargados de la venta y preparación de alimentos y bebidas. Dichos parámetros son los siguientes:

- Tiempo que tarda la app generada vs tiempo que tarda una persona en generar una cuenta de varios productos, utilizando métodos tradicionales.
- Costos monetarios generados por cometer errores en el cálculo de las cuentas.
- Encuesta de satisfacción del usuario de la app.
- Medios de distribución de la app, para darla a conocer a posibles clientes.

9.1 Cuadro de comparación de tiempos “Humano vs App”.

A continuación, en la tabla 11 se muestra la comparación de tiempo promedio utilizado para generar los subtotales y total general de la cuenta del cliente “X”. Por una parte, se muestra el tiempo utilizado la app, mientras que en segunda instancia se muestran los tiempos utilizados en hacer la misma cuenta con métodos tradicionales. Es importante mencionar que por cada método se hicieron 100 pruebas.

Tabla 11. Humano vs App (tiempo).

	App	Humano		
Método utilizado	App “CalculaTacos”	Lápiz y papel	Mentalmente	Calculadora Estándar
Tiempo	6.132 s.	20.887 s.	15.656 s.	36.525 s.

Comparación de tiempo que toma los diferentes métodos de generación de cuentas.

De acuerdo a la tabla anterior, es posible identificar que el tiempo de respuesta de la app “CalculaTacos” es el que tiene una ventaja ante los demás métodos, ya en promedio tardó tan solo 6.132 s. mientras que el método humano más cercano (Mentalmente), demoró más del doble de tiempo. A continuación, se muestra la figura 25, en donde se puede apreciar una gráfica con los tiempos de respuesta de cada método utilizado en el experimento.

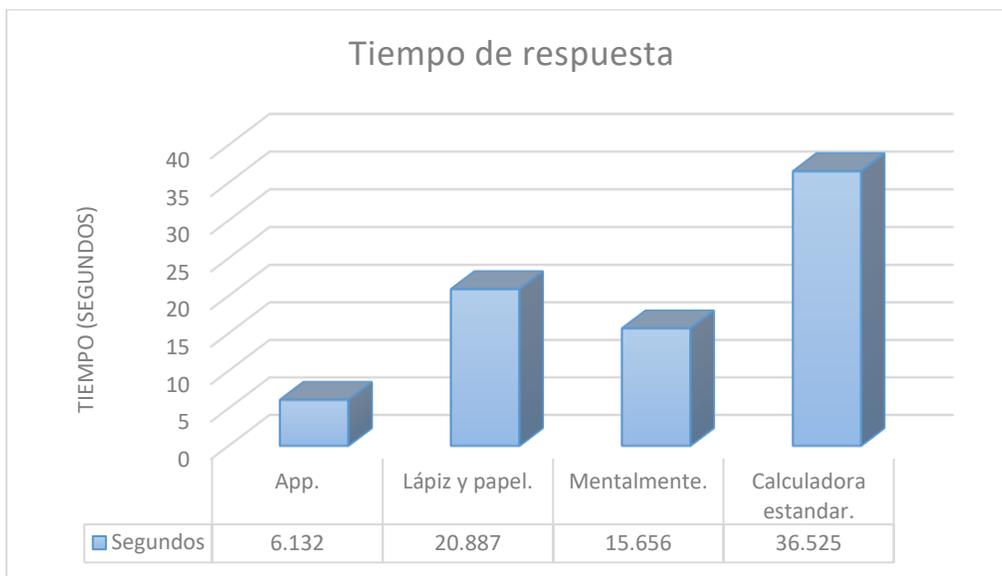


Figura 25 Tiempo de respuesta por cada método utilizado en la comparación de tiempos.

9.2 Prueba de costos

En la prueba de costos, se contaron el número de cuentas que fueron incorrectas en cada uno de los métodos. La tabla 12 refleja el número de cuentas erróneas en cada uno de los métodos utilizados.

Tabla 12. Errores cometidos en cada uno de los métodos.

	App	Humano		
Método utilizado	App “CalculaTacos”	Lápiz y papel	Mentalmente	Calculadora Estándar
Numero de errores	0	1	2	0

Cantidad de errores cometidos en la generación de cuentas mediante los diferentes métodos.

Como se puede apreciar en la tabla anterior, existió un error haciendo uso de lápiz y papel, mientras que el método mental tuvo 2 equivocaciones. El promedio monetario de las dos cuentas fue de \$118.00 MXN, En dónde el error era de \$12.00 MXN a favor de la taquería.

Aquí es importante mencionar que como las equivocaciones pueden llegar a ser constantes, repercutirá en que los clientes de este negocio se vallan con la competencia. En donde si cada cliente solía visitar a este negocio una vez por semana con un consumo promedio de \$118.00 MXN. En un mes, la perdida de ventas hablando en términos monetarios ascenderá a \$472.00 MXN. Suponiendo que fueran 5 los clientes que emulan al primero, en un mes la perdida de ventas ascendería a \$2,360.00 MXN.

Sin duda, es imperativo poner atención a la efectividad del cobro, ya que de esto depende la fidelidad de los clientes que tienen las PYMES encargadas de la preparación de alimentos y bebidas.

9.3 Encuesta de satisfacción del usuario.

Los resultados de las veinte encuestas realizadas a usuarios finales (véase anexo B) fueron realmente satisfactorios, debido a que en el 100% de las respuestas los usuarios calificaron a la app como una herramienta útil que resuelve su problemática de una forma eficaz.

Por ejemplo, a continuación, se presentan los resultados en forma de gráficas de las 3 preguntas con mayor relevancia en las encuestas. En la figura 26 se muestran los resultados de la pregunta “¿Cómo considera a la aplicación móvil “CalculaTacos” en cuanto a realizar correctamente las funciones para las cuales fue desarrollada?”. Los resultados son contundentes, porque el 80% de los encuestados califica como excelente a la app en cuanto a funcionalidad, solo existe un 20% que la califica como buena. De cualquier forma, ambas respuestas son positivas debido a que la app cumple con las funciones para las cuales fue desarrollada.

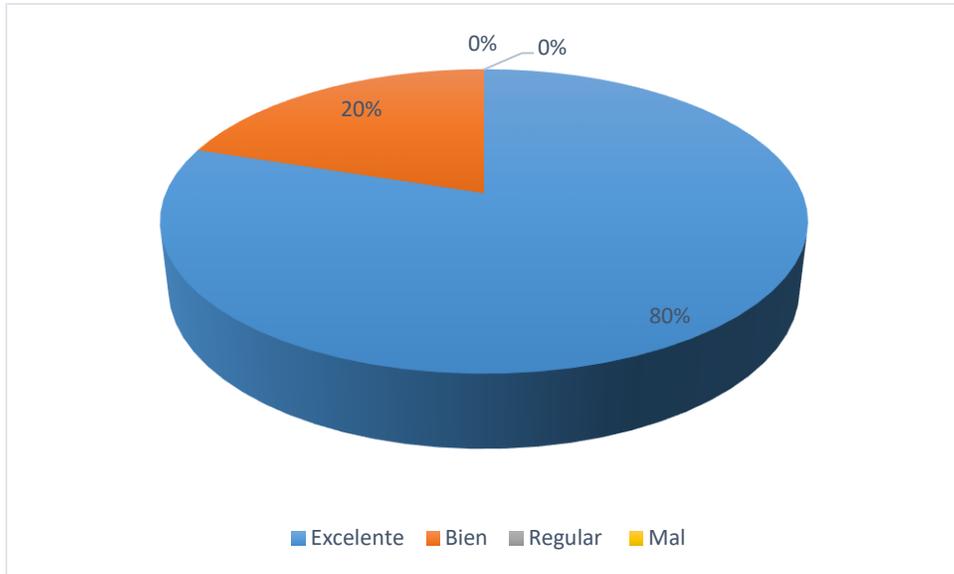


Figura 26 Aceptación de la app en cuanto a funcionalidad.

La segunda pregunta relevante es la siguiente: “¿Cuánto tiempo le tomó aprender a utilizar la app de manera correcta?”. Dicha interrogante tuvo la intención de calificar a la app en cuanto a su complejidad para hacer uso de ella. Es por ello que en la figura 27 se muestran los resultados en cuanto a la usabilidad de la app. Los resultados fueron favorecedores para la app CalculaTacos dado que el 100% de los usuarios aprendieron a utilizar la app en menos de media hora.

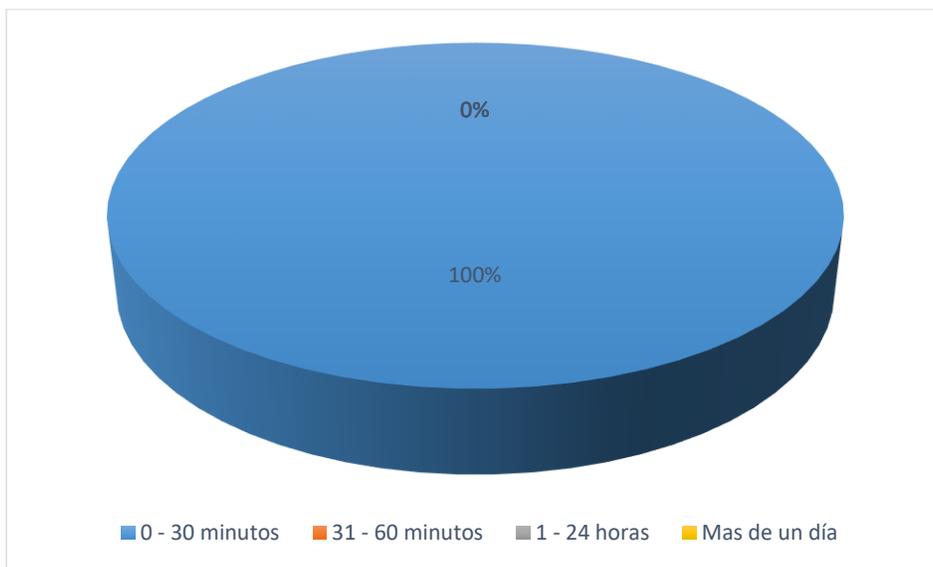


Figura 27 Aceptación de la app en cuanto a su usabilidad.

Por último, se presenta la tercera de las preguntas más relevantes de la encuesta en cuanto a la satisfacción del usuario, la cual fue la siguiente: “En el tiempo que lleva utilizando la app en su negocio, ¿Cuántas veces ha arrojado un resultado erróneo?”. Dicha interrogante tenía la finalidad de conocer el nivel de efectividad de la app. En la figura 28 se muestran los resultados de esta última pregunta. Los resultados son verdaderamente satisfactorios, puesto que el 100% de los usuarios dijeron no haber tenido ningún problema en cuanto a la aplicación. Manifestaron que la app cumple con sus funciones de una manera íntegra. Lo cual los beneficia a ellos como comerciantes en cuanto a evitar cobrar una cuenta de forma equívoca.

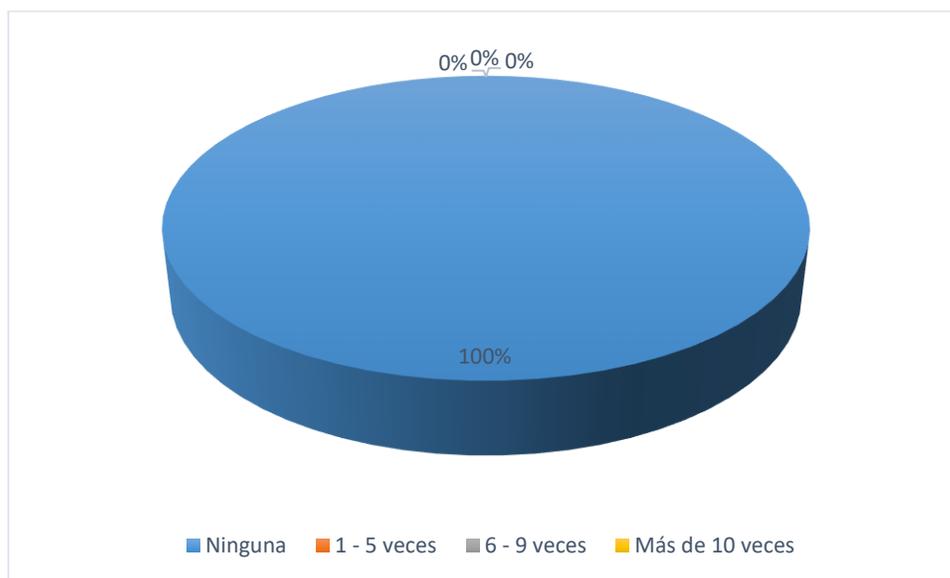


Figura 28 Nivel de efectividad de la app.

Como se pudo apreciar en las tres graficas anteriores, la app desarrollada en el presente proyecto cubre de manera satisfactoria las necesidades de los usuarios. Permitiéndoles realizar el cobro de las cuentas generadas por las ventas de sus productos de una manera precisa y fácil de realizar. De alguna manera esta app funciona como un sistema estratégico debido a que les proporciona una ventaja competitiva a los negocios que hacen buen uso de ella.

9.4 Publicación de la app en el repositorio de Play Store.

Como cuarto y último parámetro en la parte de resultados se encuentra la publicación de la aplicación móvil “CalculaTacos” en un medio electrónico para su distribución. Este medio es el más importante en cuanto a publicación de apps móviles se refiere. La app generada en el presente proyecto de investigación se publicó en Play Store.

Posterior a entregar la app a los comerciantes que inicialmente la solicitaron, se consideró la opción de poder lucrar con ella. Con base a lo anterior se optó por venderla en la tienda oficial de Google. Así que actualmente se puede encontrar en Play Store buscándola con las palabras “CalculaTacos” en la barra de búsqueda o en su defecto accediendo al siguiente enlace: <https://play.google.com/store/apps/details?id=com.tesislia.pako.calculadoratacos&hl=es>.

X DISCUSIÓN

Después de terminar el desarrollo de la app CalculaTacos como proyecto de tesis para obtener el título de Licenciado en Informática Administrativa (LIA). Viendo el trabajo en retrospectiva, es interesante rescatar que el desarrollo de aplicaciones móviles para dispositivos Android es una tarea con un alto grado de dificultad, ya que engloba el uso de diferentes tecnologías, así como la interacción entre un gestor de base de datos con pocas características y la interfaz gráfica mediante un lenguaje de marcado como lo es XML. Si bien es cierto que en algunas materias que forman parte del plan curricular de la licenciatura se utiliza la programación orientada a objetos, haciendo uso del lenguaje java. Dicho plan no carece de materias en donde se aborde el uso de nuevas tecnologías para poder satisfacer las necesidades del mercado laboral actual. En este sentido es importante mencionar que el plan de estudios se encuentra rezagado en cuanto a formación de tecnologías actuales a los alumnos que cursan esta licenciatura. Es por ello que sería conveniente que las autoridades académicas de la Universidad Autónoma del Estado de México tomen la responsabilidad de actualizar los temarios para que en un futuro cercano la universidad forme alumnos con las competencias necesarias para introducirse al mundo laboral.

En el caso práctico la app CalculaTacos sirvió a una taquería, dado que este tipo de negocios es muy común en nuestro país, y como lo demostraron las pruebas de tiempo, costos y satisfacción del usuario, cumple perfectamente con las tareas para las cuales fue creada. Sin embargo, es imperativo mencionar que con unos pequeños ajustes la app puede ser utilizada en cualquier PYME encargada de la preparación y venta de alimentos que tenga la necesidad de automatizar su proceso de ventas.

Actualmente la app CalculaTacos está disponible en la tienda oficial de Google (PlayStore). La finalidad de elegir este medio electrónico de distribución fue para poder llegar a la mayor cantidad de usuarios posible, puesto que dicha tienda cuenta con el mayor prestigio de descarga de apps. Así mismo, con una estrategia de marketing a través de redes sociales es posible darle una mayor difusión a la app con el fin de que la app pueda ponerse a la venta y así tener una manera de ingresos económicos pasivos. Dichos recursos tendrían la finalidad de ser la inversión para en un futuro cercano poder desarrollar un conjunto de apps que

ayuden a automatizar diferentes tareas a las PYMES, quienes son las que necesitan ser más competitivas.

XI CONCLUSIÓN

De acuerdo a una encuesta realizada por (Kinvey, 2016), acerca de cuánto tiempo se necesita para construir una aplicación móvil, se obtuvo que normalmente se necesitan cerca de 18 semanas de desarrollo para tener una app pulida y funcional; en donde se utilizan diferentes metodologías de desarrollo. Esto incluye el back end que es la parte de gestión de base de datos, notificaciones, etcétera; en otras palabras, es la parte no visible de una app. En estas semanas también se incluye la parte de Front end, que se refiere básicamente a la parte gráfica de una app. Tomando como referencia la estadística anterior, es importante llevar a cabo una comparación entre el tiempo promedio de desarrollo de una app (18 semanas) y el tiempo que se necesitó para terminar la aplicación “CalculaTacos” (9 semanas). Entonces, se puede concebir que al desarrollar esta app bajo una metodología de desarrollo ágil como lo es Scrum se ahorró un 50% de tiempo en comparación con el promedio.

La app fue desarrollada con el fin de ser útil a cualquier PYME que se encargue de la preparación y venta de alimentos y bebidas, sin embargo en el caso práctico para la presente tesis se tomó como ejemplo una taquería, dado que este es un negocio muy representativo de este tipo PYMES.

Por otra parte, haciendo la comparación del tiempo utilizado en realizar el cálculo de una cuenta de ejemplo bajo métodos tradicionales y empleando a la app “CalculaTacos”, se encontró que, al utilizar esta app en la cuenta de ejemplo se tomó 6.132 segundos, mientras que al utilizar el método tradicional más rápido (mentalmente), se tomó 15.656 segundos. Entonces se puede concluir que se hace un ahorro promedio del 60.833% al utilizar la app “CalculaTacos”. Por otra parte, es importante tomar en cuenta que el método tradicional más rápido fue realizando la cuenta mentalmente por una persona que tiene experiencia, sin embargo, es probable que este método llegue a fallar en determinados casos o que una persona promedio tarde más tiempo en generar ese cálculo. Así que de cualquier forma la app “CalculaTacos” sale bien librada en el parámetro de tiempo.

Otro aspecto importante a retomar, es que la app tiene un alto grado de efectividad al utilizarla correctamente, esto asegura que siempre arroje resultados verídicos. La ventaja de tener que genere las cuentas de cada una de las ventas de forma correcta es que se evita la pérdida de

clientes al momento de cobrarles de más; y así evitar que se vallan con la competencia. Este fenómeno repercute en altas pérdidas monetarias a largo plazo.

Por último, es importante señalar que, al realizar la encuesta de satisfacción a los usuarios de la app, se encontró que el grado de satisfacción está por arriba del 95%. Lo cual habla de que la app fue realizada correctamente y es funcional para los usuarios, además de que su grado de usabilidad es aceptable. Pero aún más importante, esta app resuelve el problema de cálculo y cobro de cuentas, que es uno muy común entre las PYMES de este giro. Y de alguna manera la app ayuda a ser más competitivos a los negocios que hagan uso de ella.

XII BIBLIOGRAFÍA

- Hopcroft, J., Motwani, R., & Ullman, J. (2008). Teoría de autómatas, lenguajes y computación. Pearson.
- 24, I. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
<https://play.google.com/store/apps/details?id=com.andromo.dev597108.app690294>
- Alacarta. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
<https://alacartamexico.com/>
- Almazán Morales, J. L., Méndez Salvatorio, A., & Lee kim, H. S. (2013). ¿UNA HERRAMIENTA INNOVATIVA PARA LAS PYMES? INCEPTUM, 85-109.
- Appstaurante. (07 de Febrero de 2018). Appstaurante. Obtenido de Appstaurante:
<http://www.appstaurante.com>
- Bautista, G. B. (s.f.). Clasificación de las empresas en México. Obtenido de Universidad Autónoma del Estado de Hidalgo: <https://www.uaeh.edu.mx/scige/boletin/prepa4/n5/m14.html>
- Bellman, R. (1978). Functional equations in dynamic programming.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). Unified Modeling Language. Madrid, España.
- Bucanek, J. (2009). Model-View-Controller Pattern. Apress.
- C. B., & G. K. (2005). Hibernate in action. Greenwich: Manning.
- Capó Vicedo, J., Exposito, Lanja, M., & Mesia Buades, E. (2007). La importancia de los clusters para la competitividad de las PYME en una economía global. Eure, 119-133.
- Cheung, B. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
<https://play.google.com/store/apps/details?id=com.bong.BillCalculator3>
- Chomsky, G. (1956). Chomsky y los lenguajes formales. Ochoa.
- Cleri, C. (2014). El libro de las PyMEs. Granica.
- COTEC. (1999). Pautas metodológicas en la gestión de la tecnología y la innovación para empresas. Fundación COTEC.
- Cruz, F., & López Mortero, G. (2007). Una visión general del m-learning y su proceso de adopción en el es- quema educativo. Coloquio internacional Tendencias Actuales de Cómputo e Informática en México.
- D. R., & C. R. (2012). Programación en android. España: Aula Mentor.
- Domínguez. (2009). Innovación y competitividad de PYMES, la experiencia internaciona. Benemérita Universidad Autónoma de Puebla.
- Elmasri, R., & Navathe, S. (2007). Fundamentos de sistemas de bases de datos, Quinta Edición. Pearson.

- Financiero, E. (12 de Febrero de 2014). El Financiero. Obtenido de El Financiero:
<http://www.elfinanciero.com.mx/empresas/en-mexico-solo-de-las-pymes-usan-tecnologias-de-la-informacion.html>
- G. A., & M. O. (2010). *The Definitive Guide to SQLite*, Second Edition. New York: Apress.
- Garnica González, J., & Nuño de la Parra, J. P. (2011). Una visión de la innovación como elemento clave para mejorar la competitividad en las PyMEs mexicanas. *Red Internacional de Investigadores en Competitividad*.
- Google. (20 de Marzo de 2018). Google Cloud. Obtenido de Google Cloud:
<https://cloud.google.com/speech-to-text/docs/basics>
- Grune, D., & Jacobs, C. (2008). *Parcing Techniques*. New York: Springer.
- GRUPO, I. S. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Jornadas de Ingeniería del Software y Bases de Datos.
- Gutiérrez Suarez, C. (2017). *Google play game services*. Madrid, España.
- indalquiler. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
https://play.google.com/store/apps/details?id=com.mobincube.appdomicilio.sc_HWB3XU
- INEGI. (2014). *Censos económicos 2014*.
- INEGI. (2016). *Censos económicos 2014, La industria restaurantera en México*. Aguascalientes, México.: INEGI.
- J. K. (2010). *Usung SQLite*. Sebastopol, CA: O'Reilly.
- Joyanes, L., & Zahonero, I. (2011). *Programación en Java*. Mc Graw Hill.
- Joyanes, L., Rodriguez, L., & Fernández, M. (1996). *Fundamentos de Programación, Libro de problemas*. Mc Graw Hill.
- Kenneth, R. (2012). *Essential Scrum*. Addison-Wesley.
- Kinvey. (2016). How long does it take to build a mobile app? Obtenido de
<https://www.kinvey.com/how-long-to-build-an-app-infographic/>
- Kohen karen, D., & Asín Lares, E. (2009). *Tecnologías de información en los negocios*. México, DF: Mc Graw Hill.
- Kurzwil, R. (1990). *The age of intelligent machines* . Cambridge: MIT Press.
- Letelier, P. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
- López, F. (2008). *El Lenguaje Objective-C para programadores C++ y Java*.
- Maldonado Guzmán, G., Martínez Serna, M., Garcia Pérez, D., Aguilera Enríquez, L., & González Adame, M. (2010). La influencia de las TICs en el rendimiento de la PyME de Aguascalientes. *Investigación y Ciencia de la Universidad Autónoma de Aguascalientes*, 57-65.

- Mannino, M. (2007). Administración de bases de datos, diseño y desarrollo de aplicaciones. Mc Graw Hill.
- McLaughlin, B. (2002). Building Java Enterprise Applications, Vol I: Architecture. O Reilly.
- Moreno Aroca, A. M. (2014). USO DE TECNOLOGÍAS DE INFORMACIÓN Y SU INCIDENCIA EN EL CRECIMIENTO DE LAS PYMES DEL SECTOR TEXTIL DE GUAYAQUIL. Guayaquil, Ecuador.
- N. B. (2013). Android 4 Principios de desarrollo de aplicaciones Java. España: ENI Ediciones.
- Narvin. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
<https://play.google.com/store/apps/details?id=com.narvin.android.easytaxcalculator>
- Oropeza, L., & Suárez, S. (2006). Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. Computación y sistemas, 270-286.
- Oz, E. (2008). Administración de los sistemas de la información. México, DF: CENGAGE Learning.
- Pantaleo, G., & Ludmila, R. (2014). INGENIERIA DE SOFTWARE. Alfaomega.
- Porter, M. (1990). The Competitive Advantage of Nations. New York: The Free Press.
- Pressman, R. (2010). Ingeniería del software, Un enfoque práctico. Mc Graw Hill.
- Pressman, R. (2010). Ingeniería del Software, un enfoque práctico. México DF: Mc Graw Hill.
- Proyectos Agiles. (26 de Julio de 2017). Proyectos Agiles. Obtenido de
<https://proyectosagiles.org/que-es-scrum/>
- R. S., & S. T. (2015). Mobile Learning: Nuevas realidades en el aula. OCEANO.
- Ramez, E., & Shamkant, N. (2007). Fundamentos de sistemas de bases de datos. Pearson.
- Ramos, J., Ramos, A., & Montero, F. (2006). Sistemas Gestores de Bases de Datos. Madrid: Mc Graw Hill.
- Ravenna, M. C., & González, L. (2004). Las tecnologías de la información en las pequeñas y medianas empresas mexicanas. México: Scripta Nova.
- Ravi Sethi, A., & Ullman, J. (1986). Compiladores, principios, tecnicas y herramientas.
- Reyes Bascuñana, A., & Satofimia Ruiz, J. (s.f.). Inteligencia artificial aplicada a personas con discapacidad.
- Soriano. (2007). PYMES y EMPRENDEDORES. Obtenido de
<http://www.pymesyemprendedores.com.mx/consultorio.html>
- Tanenbaum, A. S. (2009). Sistemas operativos modernos 3ra Edición. México: Pearson.
- TheFork, L. /. (07 de Febrero de 2018). Google Play. Obtenido de Google Play:
<https://play.google.com/store/apps/details?id=com.lafourchette.lafourchette>
- Wilson, L. (1993). Comparative Programming Languages. Addison-Wesley.

XIII ANEXOS

13.1 Anexo A, encuesta a PYMES encargadas de la preparación y venta de alimentos y bebidas.

ENCUESTA

Objetivo de la encuesta: Conocer la postura de los dueños de negocios encargados de la venta y preparación de alimentos acerca del cobro de pedidos en su establecimiento.

1. ¿Qué tipo de productos alimenticios vende?

- a) Alimentos. b) Bebidas. c) Ambos.

2. ¿Cuántos productos con precios diferentes comercializa?

- a) Entre 1 – 5. b) Entre 6 – 10. c) Entre 11 – 15. d) Mas de 15.

3. ¿Qué forma utiliza para calcular la cuenta a pagar por cada cliente?

- a) Mentalmente. b) Usando calculadoras estándar. c) Utilizando "Lápiz y Papel". d) Utilizando algún sistema informático.

4. ¿Cuántas personas laboran en su negocio?

- a) Entre 1 – 3. b) Entre 4 – 6. c) Entre 7 – 9. d) Mas de 10.

5. ¿En su negocio tienen bien definidos sus roles laborales y los respetan a diario?

- a) Sí. b) No.

6. ¿Qué tan frecuente se cometen errores aritméticos al momento de generar las cuentas de los clientes en horarios pico?

- a) Nunca. b) Algunas veces. c) Casi siempre. d) Siempre.

7. ¿Conoce algún sistema de información que automatice el cobro de los pedidos?

- a) Sí. b) No.

8. ¿Qué manera se le facilitaría más utilizar un sistema que automatice el cobro de pedidos en su negocio?

- a) A través del teclado. b) Por reconocimiento de voz.

9. ¿Cuánto tiempo tarda en promedio en generar una cuenta total para un cliente?

- a) 0 – 15 segundos. b) 11 – 20 segundos. c) 21 – 40 segundos. d) Mas de 41 segundos.

10. ¿Estaría dispuesto a utilizar algún sistema en su negocio para automatizar el cobro de pedidos?

- a) Sí. b) No. c) Tal vez.

11. ¿En promedio cuántos clientes recibe usted en su establecimiento diariamente?

- a) 0 – 20 b) 21 - 50 c) 51 – 100 d) Más de 100

13.2 Anexo B, encuesta de satisfacción de usuarios.

Encuesta de satisfacción

1. ¿Cómo considera a la aplicación móvil “CalculaTacos” en cuanto a realizar correctamente las funciones para las cuales fue desarrollada?

- A) Excelente B) Bien C) Regular D) Mal

2. ¿Como calificaría el diseño y la comodidad de uso de la app “CalculaTacos”?

- A) Excelente B) Bien C) Regular D) Mal

3. ¿Cuánto tiempo le tomó aprender a utilizar la app de manera correcta?

- A) 0 – 30 minutos B) 31 – 60 minutos C) 1 – 24 horas D) Más de un día

4. ¿Considera que la app realiza el cálculo de cuentas en su negocio más rápido que una persona?

- A) Si B) No C) Igual

5. ¿Cuánto tiempo lleva utilizando la app en su negocio?

- A) 1 – 7 días B) 8 – 15 días C) 16 – 30 días D) Más de un mes.

6. En el tiempo que lleva utilizando la app en su negocio, ¿Cuántas veces ha arrojado un resultado erróneo?

- A) Ninguna B) 1 - 5 veces C) 6 – 9 veces D) Más de 10 veces

7. ¿Que tipo de problemas ha tenido en cuanto al uso de la app?

- A) Resultados aritméticos erróneos B) Cierre de app inesperado C) Problemas de interfaz grafica D) Ninguno

8. ¿Considera que la app podría utilizarse en negocios de diferentes giros?

- A) Si B) No