



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO
DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL
RÉGIMEN FEL EN GUATEMALA**

Kevin Barrientos Castellanos

Asesorado por el Ing. Edgar Santos

Guatemala, noviembre de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO
DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL
RÉGIMEN FEL EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

KEVIN BARRIENTOS CASTELLANOS

ASESORADO POR EL ING. EDGAR SANTOS

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renado Escobedo Martinez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Christian Moisés De La Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. William Estuardo Escobar Argueta
EXAMINADOR	Ing. Byron Adolfo Zepeda Arévalo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL RÉGIMEN FEL EN GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 1 agosto de 2019



Kevin Barrientos Castellanos

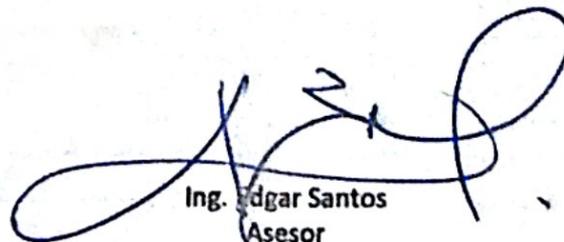
Guatemala, 4 de abril de 2020

Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Revisor de Trabajos de Graduación
Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Ingeniero Azurdia:
Tengo el agrado de dirigirme a usted para informarle que he revisado el trabajo de graduación **"DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL RÉGIMEN FEL EN GUATEMALA"**, realizado por el estudiante universitario **Kevin Barrientos Castellanos**, quien contó con la asesoría del suscrito. Considero que el trabajo realizado por el estudiante, cumple con los objetivos bajo los cuales fue planteado y cumple satisfactoriamente cada una de las actividades planificadas, por lo que procedo a aprobarlo.

Agradecimiento la atención dada a la presente,

Atentamente,



Ing. Edgar Santos
Asesor
Colegiado 5266

Edgar Santos
INGENIERO EN CIENCIAS Y SISTEMAS
Colegiado 5266



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 17 de abril de 2020

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **KEVIN BARRIENTOS CASTELLANOS** con carné 201212631 y CUI 2378 85778 0101 titulado **"DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL RÉGIMEN FEL EN GUATEMALA"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL RÉGIMEN FEL EN GUATEMALA”**, realizado por el estudiante, KEVIN BARRIENTOS CASTELLANOS aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

A handwritten signature in black ink over a circular official stamp. The stamp contains the text "UNIVERSIDAD DE SAN CARLOS DE GUATEMALA" and "DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS".

Msc. Carlos Gustavo Alonzo
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 10 de noviembre 2020

DTG. 411.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DESARROLLO DE UN PUNTO DE SERVICIO PARA LA GESTIÓN DE ÓRDENES DENTRO DE LAS PYMES DEDICADAS A LA VENTA DE ALIMENTOS Y BEBIDAS, INTEGRADA AL RÉGIMEN FEL EN GUATEMALA**, presentado por el estudiante universitario: **Kevin Barrientos Castellanos**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2020

AACE/asga

ACTO QUE DEDICO A:

- Dios** Por brindarme la capacidad, intelecto y creatividad necesaria para accionar en esta vida de manera consciente y consecuente.
- Mis padres** José Gilberto Barrientos y Ana Patricia Castellanos. Por brindarme a través de su esfuerzo y dedicación las mejores herramientas y oportunidades para crecer como una persona integral, dándome en todo momento su apoyo y cariño incondicional.
- Mis hermanos** Bryant, Kristhell y Jeffrey Barrientos. Por ayudarme a mantenerme enfocado en el cumplimiento de mis objetivos.

AGRADECIMIENTOS A:

- Universidad de San Carlos de Guatemala** *Alma máter* que por medio de su educación y exigencia me ha transformado en un profesional integro y capaz.
- Facultad de Ingeniería** Por todos aquellos catedráticos, docentes, personal administrativo e incluso de mantenimiento que pone a nuestra disposición para guiarnos en este desarrollo académico.
- Los señores** Fernando Samayoá y Anneliese Ruano. Por siempre exigir la mejor versión de mí persona.
- Mis amigos** Por ser una fuente constante de apoyo. Por todas las experiencias que hemos y seguiremos compartiendo.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. ANTECEDENTES	1
1.1. El reto de la innovación	3
1.2. Importancia de un punto de venta dentro de un negocio.....	4
2. MARCO TEÓRICO.....	9
2.1. Micro, pequeña y mediana empresa (MIPYME).....	9
2.2. Punto de venta	11
2.2.1. Puntos de venta basados en la nube.....	12
2.3. Factura electrónica en línea	13
2.4. Aplicaciones móviles	15
2.4.1. Sistemas operativos	16
2.4.1.1. Android	17
2.4.1.2. IOS	18
2.4.2. Diferencias entre una aplicación web y una aplicación móvil	19
2.4.3. Tipo de aplicaciones según su desarrollo.....	20
2.4.3.1. Aplicaciones nativas	20
2.4.3.2. Aplicaciones híbridas.....	21

2.4.3.3.	Ventajas y desventajas del desarrollo de una aplicación híbrida vs. una nativa.....	22
2.4.4.	Progressive web apps	23
3.	BENCH MARK	25
3.1.	Loyverse.....	27
3.2.	Kyte.....	29
4.	DISEÑO DE LA APLICACIÓN	33
4.1.	Prototipo.....	34
4.1.1.	Autenticación.....	34
4.1.2.	Órdenes.....	35
4.1.2.1.	Lista de órdenes.....	36
4.1.2.2.	Creación y edición de una orden.....	39
4.1.2.3.	Despliegue de orden	42
4.1.3.	Menú de platillos	43
4.1.4.	Plato	44
4.1.4.1.	Formulario para la creación y edición de un plato.....	46
4.1.4.2.	Eliminación de un plato	48
4.1.5.	Factura electrónica en línea (FEL)	50
4.1.6.	Parametrización del sistema	52
4.2.	Diseño intuitivo y usabilidad	53
4.2.1.	Paleta de colores.....	54
4.2.2.	Tipografía	56
4.2.3.	Navegación	57
4.3.	Arquitectura del software.....	58
4.3.1.	Clean architecture	59

4.3.1.1.	Arquitectura de 3 capas	60
4.3.1.1.1.	Capa de presentación: MVC	61
4.3.1.1.2.	Capa de dominio	62
4.3.1.1.3.	Capa de datos: <i>repository pattern</i>	63
4.3.2.	Flujo de datos	64
4.4.	Arquitectura del sistema	65
4.4.1.	Aplicación móvil	66
5.	DOCUMENTACIÓN	69
5.1.	Ionic	69
5.2.	Angular	70
5.3.	Firebase	72
5.4.	Cloud Firestore	72
5.5.	Firebase authentication	73
	CONCLUSIONES	77
	RECOMENDACIONES	78
	BIBLIOGRAFÍA	79

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Población según sector ocupacional	2
2.	Costos de venta de alimentos por tipo de operador.	6
3.	Áreas generadoras de pérdida	7
4.	Proceso FACE vs FE	14
5.	Marketshare por Sistema Operativo en los últimos años	17
6.	Arquitectura interna de una Aplicación Híbrida	22
7.	Loyverse IU	28
8.	Kyte IU	30
9.	Pantalla de autenticación	35
10.	Pantalla de órdenes	37
11.	Opciones de visualización de órdenes	38
12.	Opciones por orden	39
13.	Formulario de orden	41
14.	Campo emergente para pedidos a domicilio	41
15.	Detalle de orden	43
16.	Menú	44
17.	Formulario para agregar un plato a la orden	45
18.	Formulario de un plato	47
19.	Formulario para agregar una opción de preparación	48
20.	Confirmación para la eliminación de un plato	49
21.	Datos del contribuyente	50
22.	Orden ya facturada	51
23.	Pantalla de configuración del sistema	52

24.	Configuración de los datos del emisor	53
25.	Ejemplo de la paleta de colores	55
26.	Sistema de tamaños de fuente	57
27.	Navegación	58
28.	Diagrama de la arquitectura	60
29.	Capas de la arquitectura	61
30.	Patrón de diseño MVC	62
31.	Diagrama de la capa de domino	63
32.	Repository pattern	64
33.	Flujo de datos	65
34.	Módulos principales de la arquitectura	67
35.	Flujo de comunicación entre la aplicación móvil y Firebase	68

TABLAS

I.	Criterios para la clasificación de MIPYMES en Guatemala	10
II.	Criterios de evaluación del benchmark	26
III.	Métodos de autenticación Firestore	74

GLOSARIO

AGEXPORT	Asociación Guatemalteca de Exportadores, institución privada sin fines de lucro que busca promover el crecimiento de las exportaciones en Guatemala.
Android	Sistema operativo mayormente desarrollado por Google.
Angular	Framework de desarrollo que provee una arquitectura y un modelo de desarrollo base para la elaboración de aplicaciones web, comúnmente SPAs.
Assets	Recursos web que se cargan al navegador para cuando carga una página web.
API	Application programming interface, es un conjunto de funciones y rutinas que proveen las librerías en programación.
Base code	Código fuente de una aplicación reutilizado para la generación de paquetes instalables en más de una plataforma.
Base de datos	Conjunto de datos relacionados entre si y guardados sistemáticamente para su resguardo y fácil acceso.

BCIE	Banco Centroamericano de Integración Económica, el cual busca promover la integración y el desarrollo económico y social equilibrado de la región centroamericana
Certificador autorizado	Persona individual o jurídica autorizada por la SAT para encargarse de certificación de los Documentos Tributarios Electrónicos.
CIG	Cámara de la Industria Guatemalteca, organización que fomenta el desarrollo de las industrias en Guatemala para el crecimiento económico.
CONCYT	Consejo Nacional de Ciencia y Tecnología, órgano de dirección del SINCYT.
CSS	Cascade stylesheets, tiene que ver con el diseño gráfico de los sitios web dentro de HTML.
DOM	Document object model, representa un formato estándar y su interfaz del modelado de páginas web.
DTE	Documento Tributario Electrónico.
FACE	Régimen anterior de Factura Electrónica.
FEL	Factura Electrónica en Línea, es el nuevo régimen para la generación de una factura electrónica según acuerdo directorio número 13-2018.

FEPYME	Federación de la Pequeña y Mediana Empresa, organización que vela por los intereses de las pequeñas y medianas empresas asociadas, representándolos ante las entidades públicas y privadas, nacionales e internacionales.
Framework	Se refiere a una solución que provee un conjunto de herramientas para facilitar el desarrollo de aplicaciones.
GPS	Global Position System, este nombre corresponde al dispositivo que detecta la posición del usuario por medio de coordenadas.
Hardware	Componentes físicos de un computador.
Heterogeneidad	Mezcla de partes de diversa naturaleza en un todo.
HTML	HyperText markup language, se refiere a un lenguaje de marcado por medio de tags, especial para crear páginas web.
HTTP	<i>HyperText Transfer Protocol</i> , protocolo de comunicación usado en la Web.
Ionic	<i>Framework</i> de código abierto que provee un set de herramientas para desarrollar la interfaz de aplicaciones móviles y de escritorio usando tecnologías web como HTML, CSS y JavaScript.

IOS	Sistema operativo desarrollado por Apple.
IU	Interfaz de usuario.
JavaScript	Lenguaje de programación mayormente usado para el desarrollo de aplicaciones web.
La Nube	La computación en la nube se refiere a una manera de proveer servicios por medio de una red; usualmente la web.
Marketshare	Porcentaje de las ganancias dentro de una industria que corresponden a una compañía en específico.
Material Design	Guía técnica propuesta por Google para el desarrollo de aplicaciones con el fin de estandarizar la experiencia de usuario en las diferentes plataformas.
MINECO	Ministerio de Economía en Guatemala.
MIPYME	Micro, pequeña y mediana empresa.
MVC	Modelo Vista Controlador, es un patrón de diseño para el desarrollo de <i>software</i> .
MVP	Modelo Vista Presentador, es un patrón de diseño de <i>software</i> .

MVVM	Modelo Vista Vista-Modelo, es un patrón de diseño de <i>software</i> .
PaaS	<i>Platform as a Service</i> , es un modelo para la entrega de <i>software</i> donde el dueño de la aplicación provee la configuración necesaria para que su solución se publique en la nube.
PIB	Producto interno bruto, medida macroeconómica que expresa el valor monetario de la producción de bienes y servicios de un país.
Plug-in	Componente independiente de <i>software</i> que provee un servicio en específico. Estos componentes se adscriben a una solución más completa que consume los servicios expuestos.
PWA	<i>Progressive Web App</i> , se refiere a una aplicación web capaz de instalarse en la pantalla de inicio (<i>homescreen</i>), trabajar offline y mantener procesos en segundo plano, dando la impresión de ser una aplicación nativa.
PWA Manifest	Archivo de configuración en formato JSON que provee la información necesaria para que la plataforma o SO pueda presentar la aplicación al usuario y pueda interactuar con la misma correspondientemente.

PYMES	Pequeñas y medianas empresas, según el número de empleados, ventas o activos.
SaaS	<i>Software as a Service</i> es un modelo de licenciamiento y entrega de servicios por medio de un sistema alojado centralmente en la nube.
SAT	La Superintendencia de Administración Tributaria es la entidad encargada de la administración tributaria y aduanera en Guatemala.
SDK	Software Development Kit: conjunto de herramientas dentro de un paquete de software que ayuda a el desarrollo de aplicaciones.
SENACYT	Secretaría Nacional de Ciencia y Tecnología, órgano coordinador del SINCYT.
Serverless	Es un modelo de entrega de soluciones en la nube donde el proveedor de la nube ejecuta el servidor y administra dinámicamente la asignación de recursos de la máquina.
SINCYT	Sistema Nacional de Ciencia y Tecnología, institución que busca promover el desarrollo científico y tecnológico en Guatemala.
SPA	<i>Single Page Application</i> , es una aplicación web que no requiere recargar sus <i>assets</i> para la actualización de sus pantallas.

Testable

Característica de una solución que puede ser validada mediante diferentes mecanismos de pruebas.

POS

Point of service.

RESUMEN

En Guatemala, existe un gran índice de emprendimiento, sin embargo, la mayor parte de los negocios permanecen como pequeñas o medianas empresas. Dentro de la ciudad de Guatemala, se encuentran un sinnúmero de restaurantes que reflejan un pobre manejo en la recepción y cobro de sus órdenes. Esto limita su capacidad de atención y brinda un mal servicio al cliente. Por ello, se pretende desarrollar un punto de venta capaz de adaptar los procesos actuales de estas PYMES al mismo tiempo que genera un flujo claro en los mismos. De esta manera, se espera mitigar los problemas de comunicación entre áreas y mejorar el control de las ventas. El punto de venta deberá ser ligero, intuitivo, de fácil instalación y con una arquitectura económica, de tal forma que su desarrollo no dure más de tres meses.

OBJETIVOS

General

Proveer una solución tecnológica para implementar un punto de venta dentro de las PYMES dedicadas a la venta de alimentos y bebidas, que cumpla con los lineamientos establecidos por el régimen de factura electrónica (FEL) en Guatemala para posteriormente integrarse a un certificador autorizado.

Específicos

1. Desarrollar un punto de venta en una aplicación móvil que posea una interfaz intuitiva y moderna.
2. Desarrollar un punto de venta cuya infraestructura sea económica y fácil de implementar.
3. Desarrollar un estudio de mercado (*benchmarking*) para la solución propuesta.
4. Integrar el punto de venta con un certificador autorizado por la SAT para la generación de una factura electrónica en línea.

INTRODUCCIÓN

Guatemala es un país en vías de desarrollo cuya cultura se ha adaptado paulatinamente a la nueva forma de operar del mundo por medio de la tecnología. Por ello, es posible incorporar nuevos servicios que desarrollen la sociedad con la ayuda de la tecnología.

Según el Instituto Nacional de Estadística (INE), en Guatemala el 70,2 % de las personas trabaja en el sector informal. Dentro de la región urbano-metropolitano el sector informal representa un 45,7 % de la participación económica. Esto evidencia la alta cantidad de negocios que por falta de voluntad o por falta de capacidad no han logrado incorporarse formalmente a la economía del país.

Se habla de falta de capacidad para formar parte del sector formal cuando una empresa es incapaz de cumplir con las obligaciones económicas que exige la ley. Estas obligaciones incluyen, facturación de servicios, pago de impuestos, pago de prestaciones a los empleados, entre otros. Estas obligaciones no solo representan un gasto adicional en el que deben incurrir sino también representa un mayor esfuerzo en logística y en el control de sus procesos. Por eso, el acompañamiento a las PYMES es necesario para mejorar los procesos que coadyuven a su crecimiento de manera significativa.

Procesos como la implementación de un punto de venta pueden ayudar a llevar un mejor control de las ventas del negocio y sus ganancias. Al mismo tiempo, puede mejorar el control del inventario, optimizando las compras y las ganancias.

1. ANTECEDENTES

Según las proyecciones y estimaciones de las Naciones Unidas, Guatemala cuenta con una población de más de 17 millones¹ que junto con el PIB neto de 138 mil millones² hacen de su economía una de las más grandes en Centroamérica. Sin embargo, la tasa de pobreza y otros indicadores de crecimiento social siguen siendo bajos comparados con otros países de ingresos similares. La tasa de pobreza, analfabetismo y acceso a la información son áreas claves, flagelos de especial interés para el crecimiento de una nación.

Al igual que la mayor parte de los países centroamericanos, la economía de Guatemala está compuesta en gran parte por actividades de comercio informal. Para el 2017 se estimó que el 70,2 % de las personas laboran en el sector informal³, figura número 1. Así mismo, el Ministerio de Economía de Guatemala, estima que las PYMES contribuyen con el 35 % del PIB⁴ 2015. Por ello, parte de la agenda de las políticas nacionales busca mejorar el desempeño de las PYMES e incrementar su participación tanto en los mercados locales como mundiales. Medidas que aumenten la productividad, bajen los costos de operación, estimulen el acceso a nuevas tecnologías e integren a las PYMES a las cadenas de valor son parte de las actividades en las que se involucran entidades como FEPYME, el CONCYT, SENACYT, AGEXPORT, entre otras.

¹ Naciones Unidas. *Country Profiles*. <<https://esa.un.org/unpd/wup/DataQuery/>>

² Central Intelligence Agency, *The WORLD FACTBOOK*. <[https://www.cia.gov/library/publications/the-world-factbook/rankorder/2001rank.html#gt](https://www.cia.gov/library/publications/the-world-factbook/rankorder/2001rank.html#gt;)>.

³ Instituto Nacional de Estadística INE. *Encuesta Nacional de Empleo e Ingresos – ENEI, 2017*, p. 31.

⁴ Ministerio de Economía. *Sistema Nacional de Información MIPYME Guatemala, 2015*, p. 20.

Figura 1. **Población según sector ocupacional**



Fuente: Instituto Nacional de Estadística INE. *Encuesta Nacional de Empleo e Ingresos – ENEI*, 2017. p. 31.

Las PYMES en Guatemala enfrentan diferentes obstáculos económicos y culturales que limitan su competitividad e impiden su crecimiento. Un informe realizado en el 2010 por el Banco Mundial enmarca 5 ejes que deben trabajarse de manera íntegra dentro de la agenda nacional para poder apoyar a este tipo de empresas. Los retos descritos son⁵:

- *El reto de la infraestructura*
- *El reto del capital humano e innovación*
- *El reto del acceso al financiamiento*
- *Las regulaciones empresariales y la informalidad*
- *La priorización de inversiones en áreas rurales*

⁵ Banco Mundial. *Desarrollo de PyMEs en Guatemala, Facilitando el Florecimiento de 10 000 Empresas*, informe No. 54242-GT, 2010. p. 28.

Para esta investigación es importante lo discutido en el punto 2 ya que se describe un marco de referencia claro en el cual las PYMES se manejan, sus motivaciones, problemas consecuentes y posibles soluciones.

1.1. El reto de la innovación

La innovación para una empresa puede generarse desde la incorporación de una nueva tecnología hasta la actualización de productos o procesos que mejoren su calidad e incrementen su competitividad. Es el principal motor de crecimiento y la mejor herramienta que puede tener una empresa para mejorar su competitividad. En general, la innovación puede ayudar en la mejora de la calidad y el diseño de los productos, la reducción de costos, el tiempo de entrega y otros factores. Por ello, es necesario hacer que los dueños de PYMES vean en la innovación tecnológica una oportunidad clara de crecimiento para que puedan adoptar estrategias concretas para su desarrollo.

El informe presentado por el Banco Mundial determinó que las instituciones encargadas de la transferencia de tecnología, como el CONCYT y el SENACYT, carecían de un mandato sólido, capacidades y recursos suficientes. Por otro lado, se identificó que las empresas de calidad poseen un proceso sólido de acreditación, sin embargo, es lento cuando actualizan sus normativos a estándares internacionales.

Así, el Banco Mundial recomienda en el 2010 evaluar y mejorar el liderazgo, visión y estrategia de las instituciones encargadas de la transferencia de conocimiento. Guatemala debe darle mayor importancia a dicha transferencia desde los sectores más altos del gobierno, específicamente, la presidencia. Se debe mejorar la coordinación interinstitucional.

Al mismo tiempo, se deben de les debe fortalecer para mejorar y facilitar el acceso a la información y la transferencia del conocimiento. Para ello, se puede promover la instalación de Centros de Innovación Tecnológicos (CITIES) encargados de:

- transferir información sobre nuevas tecnologías
- tratar asuntos de calidad
- identificar cuellos de botella y oportunidades dentro de las PYMES y
- facilitar la comercialización de nuevos productos

Por otro lado, el Centro de Investigaciones Económicas Nacionales (CIEN), en su informe sobre los lineamientos de política económica social y de seguridad 2012-2020 enfocado a MIPYMEs también hace énfasis sobre el reto de la innovación, estableciendo a esta como una estrategia necesaria para todas las empresas. El CIEN refuerza los puntos descritos por el Banco Mundial, concluyendo que Guatemala no posee una cultura que promueva la innovación y la calidad.

1.2. Importancia de un punto de venta dentro de un negocio

En la actualidad, los puntos de venta modernos para restaurantes constituyen una plataforma que es el eje de las operaciones de transaccionales a menor y mayor escala, lo que contribuye a mejorar la experiencia de los clientes y optimizar las operaciones de negocios. Son esenciales para los operadores que busquen ofrecer una mejor experiencia a los clientes, mantener el control de sus costos y aumentar los ingresos y la rentabilidad en general.

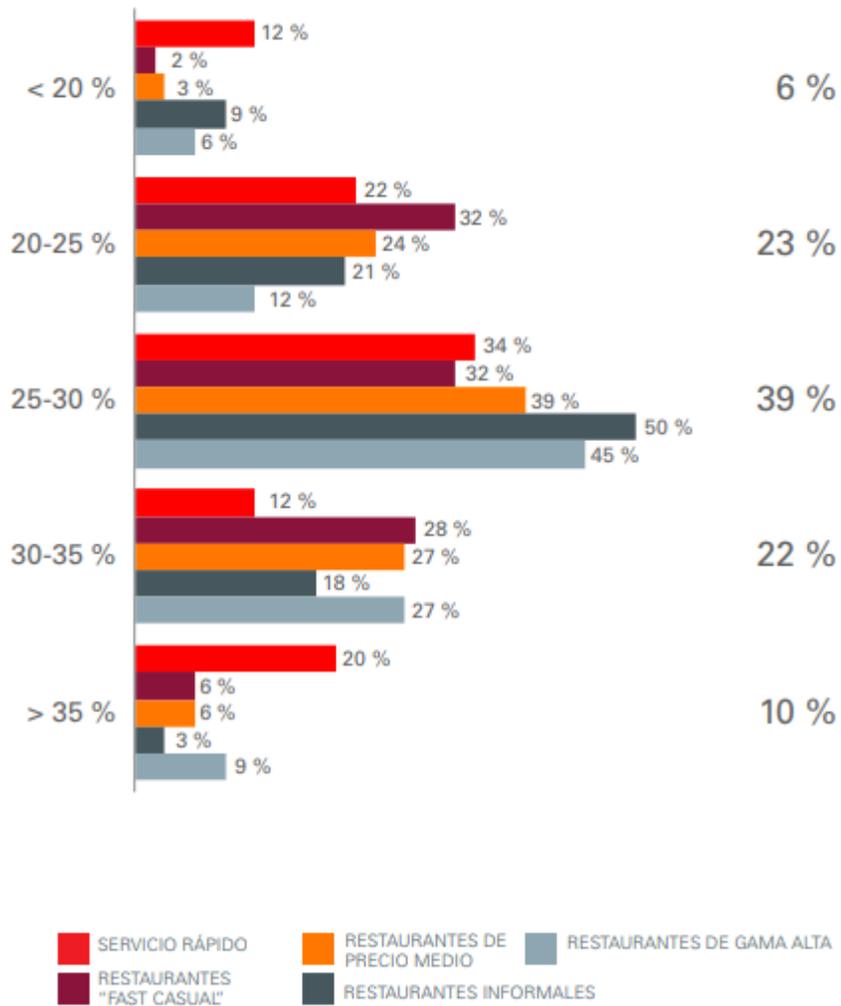
En un informe realizado por Technomic para Oracle Hospitality sobre el control de costes en el sector de la restauración se evidencia como la gestión del

personal, el inventario y la prevención de pérdidas es un reto global. El informe pretende vislumbrar la oportunidad que la tecnología ofrece para mitigar dichos problemas al integrar las soluciones correctas dentro del negocio.

Technomic determinó que el 71 % de los operadores independientes reportaron que el coste de la materia prima asciende hasta a un 25 % del total de las ventas, incluso un 10 % afirmó que el coste se elevaba a un 35 %, según se observa en la figura 2. Considerando el impacto que este coste representa, mantener el control de la materia prima resulta un factor clave para elevar los dividendos del negocio.

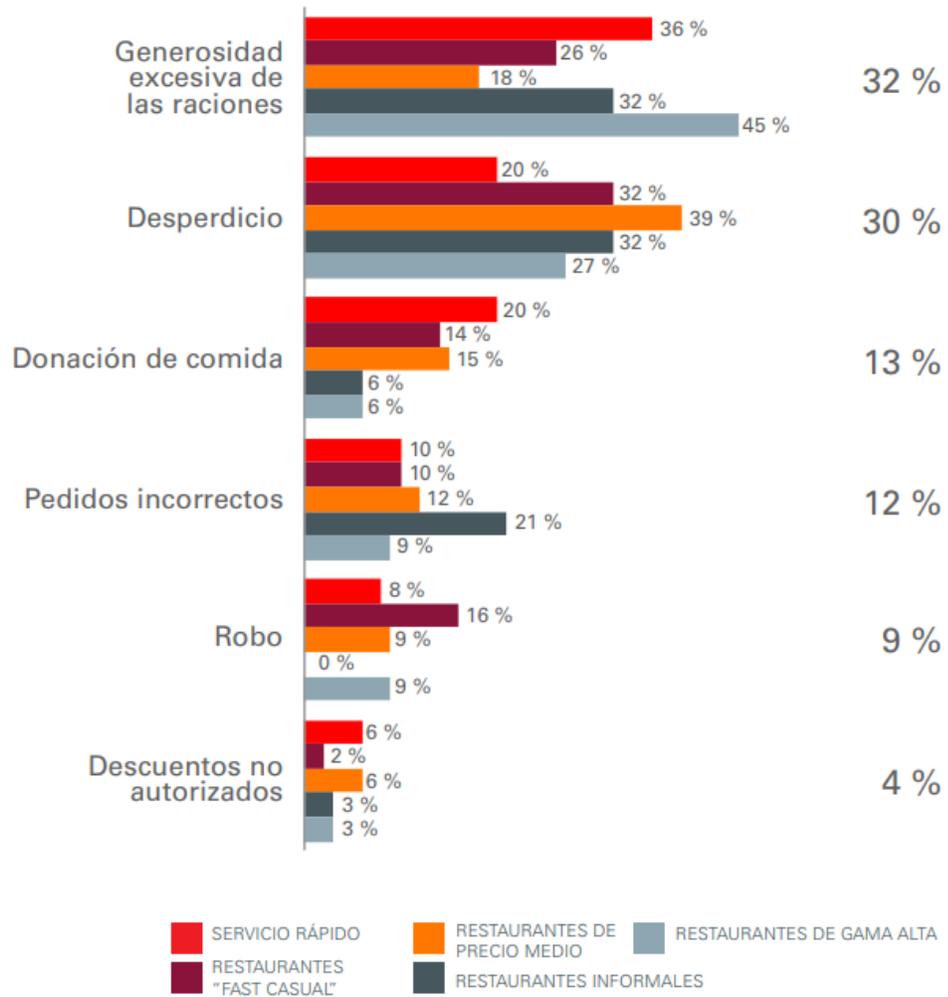
Además de los costos de materia prima Technomic determinó que las principales áreas de pérdida se debían a la excesiva generosidad de las raciones y el desperdicio de comida ver figura 3. Así mismo, existen otros factores generadores de pérdidas, como la donación de comida, pedidos incorrectos, robo y descuentos no autorizados. De esta manera, se evidencia la necesidad de un mayor control por medio de un punto de servicio.

Figura 2. **Costos de venta de alimentos por tipo de operador**



Fuente: Oracle Hospitality. *Control de costes en el sector de la restauración*. p. 11.

Figura 3. **Áreas generadoras de pérdida**



Fuente: Oracle Hospitality. *Control de costes en el sector de la restauración*. p. 12.

2. MARCO TEÓRICO

2.1. Micro, pequeña y mediana empresa (MIPYME)

El concepto y clasificación de las micro, pequeñas y medianas empresas o PYMES es una discusión abierta debido a la diversidad y heterogeneidad de las empresas que componen dichos grupos. Tanto en el ámbito internacional como en el nacional, tratar de englobar bajo un único concepto a dichas empresas resulta difícil y complejo. En general, no existen definiciones únicas, ya que cada mercado toma en cuenta sus propios criterios con base a sus condiciones.

Uno de los criterios más frecuentes considerados en los ámbitos internacional y nacional, es el número de trabajadores que emplea la empresa. En los países en desarrollo, además del número de empleados, se aplican variables, como el valor de los activos y el volumen de ventas.

Específicamente, en Guatemala, existen 4 criterios comúnmente usados para su clasificación. El Ministerio de Economía, en el Artículo 1 del Acuerdo Gubernativo 211-2015, categoriza las MIPYMES por medio del número de empleados y ventas anuales; el Banco Centroamericano de Integración Económica (BCIE) también utiliza el criterio de número de empleados, pero con diferentes dimensiones para cada segmento; la Cámara de la Industria (CIG) utiliza dos tipos de definiciones: la primera que clasifica a los beneficiarios de programas de apoyo (con un criterio de cantidad de empleados), y la otra para clasificar las empresas industriales (basado en activos totales, cantidad de empleados y ventas anuales); y, finalmente AGEXPORT, utiliza el criterio de

tamaño de activos, ver tabla I. Para fines de esta investigación se usará el criterio del MINECO.

Tabla I. **Criterios para la clasificación de MIPYMES en Guatemala**

Criterio	Usado por	Micro	Pequeña	Mediana
Número de empleados	MINECO	1 – 10	11 – 80	81 – 200
	CIG	1 – 5	6 – 50	51 – 100
	BCIE	1 – 10	11 – 40	41 – 60
Ventas	MINECO	De 1 salario mínimo hasta 190 salarios mínimos	De 191 salarios mínimos hasta 3 700 salarios mínimos	De 3 701 salarios mínimos hasta 15 420 salarios mínimos
Activos	AGEXPORT	No aplica	Menos de Q. 500 000	Hasta Q. 1 200 000
Empleados, Ventas y Activos	Empresas industriales	De 1 a 10 empleados; hasta Q. 60 000 en ventas anuales; Hasta Q. 50 000 en activos	De 11 a 20 empleados; hasta Q. 300 000 en ventas anuales; hasta Q. 500 000 en activos	De 21 a 50 empleados; hasta Q. 500 000 en ventas anuales; hasta Q. 2 000 000 en activos

Fuente: elaboración propia.

2.2. Punto de venta

Es el lugar y espacio donde se lleva a cabo la venta de un bien o servicio de un negocio. El proceso de venta para un negocio dado comúnmente involucraría el cálculo total adeudado, la preparación de una factura con los bienes por adquirir. Este punto también es donde el cliente efectuará el pago. Tras completar el pago el comerciante puede que genere un recibo y se lo entregue al cliente como comprobante del mismo, ya sea impreso o vía electrónica.

El punto de venta también es conocido como punto de servicio, pues en la actualidad este maneja más que solo el proceso de venta también maneja otros procesos de interés para el negocio. Actualmente, un punto de servicio cuenta con funcionalidades como un control de inventario, gestión de clientes, reportes financieros, por mencionar algunas de las más comunes.

Estos sistemas son de los más complejos porque los requieren todo tipo de negocio. Entre las características comunes que los negocios requieren se encuentran manejo de venta, inventario, recuento de inventario, pedidos de proveedores, fidelización de clientes y reportes varios. Así pues, estas son las funcionalidades mayormente incluidas en este tipo de sistemas. Además, se requiere que el sistema interconectado cada uno de estos módulos de manera que pueda ser práctico y funcional para el negocio.

Para un punto de venta, la interfaz de usuario es especialmente importante. Cuanto más natural, claro y rápido sea el flujo para completar una venta, mejor será para el negocio. Debido que muchos negocios suelen tener horas de mucha clientela por ello les interesa poder completar la venta lo antes posible para poder atender a más clientes. Sin embargo, en el afán de reducir el

flujo y las posibles distracciones en las que se pueda ver envuelto un vendedor, se puede terminar sacrificando funcionalidades importantes, como el manejo de descuentos, manejo de lealtad de clientes, entre otros.

2.2.1. Puntos de venta basados en la nube

La introducción y mejora de la computación en la nube han provocado el desarrollo de muchos sistemas POS tipo SaaS (*Software as a Service*), los cuales pueden ser accedidos directamente desde internet por medio de un navegador web. De esta forma, los sistemas son independientes de la plataforma y sistema operativo en el que corren. Por medio de esta infraestructura los POS se han expandido hacia plataformas móviles, como tabletas o teléfonos inteligentes. Estos últimos debido al *hardware* incorporado pueden ser usados como lectores de código de barras y lectores de tarjetas de crédito simplificando la instalación de estos.

Los sistemas POS en la nube se diferencian de los tradicionales debido a que almacenan la información del usuario (clientes, ventas, inventario, entre otros) remotamente. Esto hace que dicha información esté disponible en otras terminales con acceso a internet.

Estos sistemas a su vez suelen ofrecer actualizaciones con mayor frecuencia. Incrementando la rentabilidad para el negocio. En general, un sistema POS tipo SaaS espera reducir significativamente la inversión inicial y el esfuerzo de instalación requerido por el negocio que desee implementar dicho sistema.

2.3. Factura electrónica en línea

El FEL o la Factura Electrónica en Línea es el nuevo régimen para la generación de una factura electrónica, autorizado por la Superintendencia de Administración Tributaria (SAT) en el acuerdo de directorio número 13-2018. Este régimen reemplaza a la Factura Electrónica (FACE) con una mayor estandarización, flexibilidad y escalabilidad. La SAT pretende que todas las empresas se incorporen a este régimen de manera progresiva, empezando por las empresas más grandes y concluyendo con las PYMES, sin embargo, se puede efectuar el cambio de manera voluntaria en cualquier momento.

“Es el modelo operativo y las normas aplicables a la factura electrónica en línea, que comprende la emisión, transmisión, certificación y conservación por medios electrónicos de facturas, notas de crédito y débito, recibos y otros documentos autorizados por la Superintendencia de Administración, que se denominan Documentos Tributarios Electrónicos.”⁶

El nuevo régimen pretende solventar las fallas que el FACE presentaba para las empresas en cuanto a inscripción, flexibilidad e implementación. Además, este régimen supone varias mejoras en la estandarización de la información que conforman los Documentos Tributarios Electrónicos (DTE) haciendo posible la elaboración de sistemas capaces de integrarse con diferentes certificadores en un menor tiempo. Así mismo, el proceso de autenticación y autorización ha sido redefinido para lograr un proceso de los documentos más rápido y con menor gasto de recursos. Estos cambios se resumen en la figura 4.

⁶ Superintendencia de Administración Tributaria. *Nuevo Régimen de Factura Electrónica en Línea (Acuerdo de Directorio Número SAT-13-2018)*. <https://portal.sat.gob.gt/portal/efactura/>.

En este nuevo régimen existen cinco actores principales: la SAT, los certificadores, los auditores, los emisores y los receptores. La SAT es el ente que establece la normativa del modelo de operación bajo el cual certificadores y emisores han de registrarse. Además, funciona como certificador gratuito por medio de la agencia virtual. Los certificadores son las personas individuales o jurídicas que certifican los DTEs generados por los emisores. Estos son autorizados por la SAT tras ser evaluados por auditores que validan el cumplimiento de las normas para garantizar la seguridad de la información. Los emisores son los contribuyentes como tal que generan los DTEs. Finalmente, los receptores son los consumidores finales.

Figura 4. **FACE vs FEL (Principales diferencias)**

FACE						
Emisor debe tener BancaSAT y esperar próximo mes para iniciar con la emisión	Los GFACE únicamente pueden ser personas jurídicas	Los emisores deben optar solo por un GFACE	Aut. de Docs. es igual que en papel (número de autorización, rangos)	Modalidad de factura "pura" y resguardo electrónico del papel	GFACE almacena información y la comparte a SAT en consultas web	DTE solo llevan un Código de seguridad que generan los GFACE
1	2	3	4	5	6	7
FEL						
Emisor debe tener Agencia Virtual, e inicia la emisión de inmediato	Certificadores pueden ser personas jurídicas o individuales	Emisores pueden optar por uno o más Certificadores y la Agencia Virtual	Aut. de Docs. es en forma individual y única por cada DTE	Solo existe la factura electrónica en línea, en tiempo real	Certificadores envían a SAT de inmediato los DTE autorizados para su validación y resguardo.	Los DTE cuentan con firma electrónica tanto del emisor como del certificador

Fuente: SAT. *Régimen de factura electrónica en línea (FEL)*.

<http://ccg.com.gt/web-ccg/wp-content/uploads/2014/10/FEL-presentacio.pdf>.

Consulta: 05 de junio 2019.

2.4. Aplicaciones móviles

Las aplicaciones móviles, mejor conocidas como Apps, son programas de computadora diseñados para ser ejecutados en un dispositivo móvil, como los *smartphones* o tabletas, de manera que extiendan su funcionalidad. La necesidad de los usuarios de agregar más funcionalidad a sus dispositivos dio origen a las Apps. Actualmente, existe una infinidad de aplicaciones disponibles en las principales tiendas como lo son la Play Store y Apple App Store, sumando cerca de 6 millones de aplicaciones⁷. Así pues, es de esperar que exista más de una aplicación disponible para solucionar un mismo problema.

A pesar del gran número de aplicaciones disponibles, existen muchas ya descontinuadas u obsoletas. Tras 10 años de desarrollo de los sistemas más populares como IOS y Android cada cual ha evolucionado ofreciendo más herramientas para los desarrolladores, pero al mismo tiempo implementando nuevos estándares de diseño y desarrollo. Para tales efectos, no todas las aplicaciones han sido actualizadas. Esto provoca una fragmentación de las soluciones disponibles, por eso, algunas son incompatibles, otras inoperables por su pobre interfaz o uso de las herramientas. Por ello, es necesario que surjan nuevos desarrollos para proveer al usuario una solución que utilice correctamente y explote al máximo el hardware disponible por medio de la tecnología ofrecida por el sistema operativo, pero que, a su vez, cumpla con los estándares de diseño propuestos para cada sistema. Así el usuario final podrá disfrutar de ambos mundos, una aplicación funcional y atractiva.

En las siguientes secciones se describen conceptos fundamentales para el desarrollo de una aplicación móvil desde los sistemas operativos disponibles,

⁷ Statista. *Number of apps available in leading app stores.* <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.

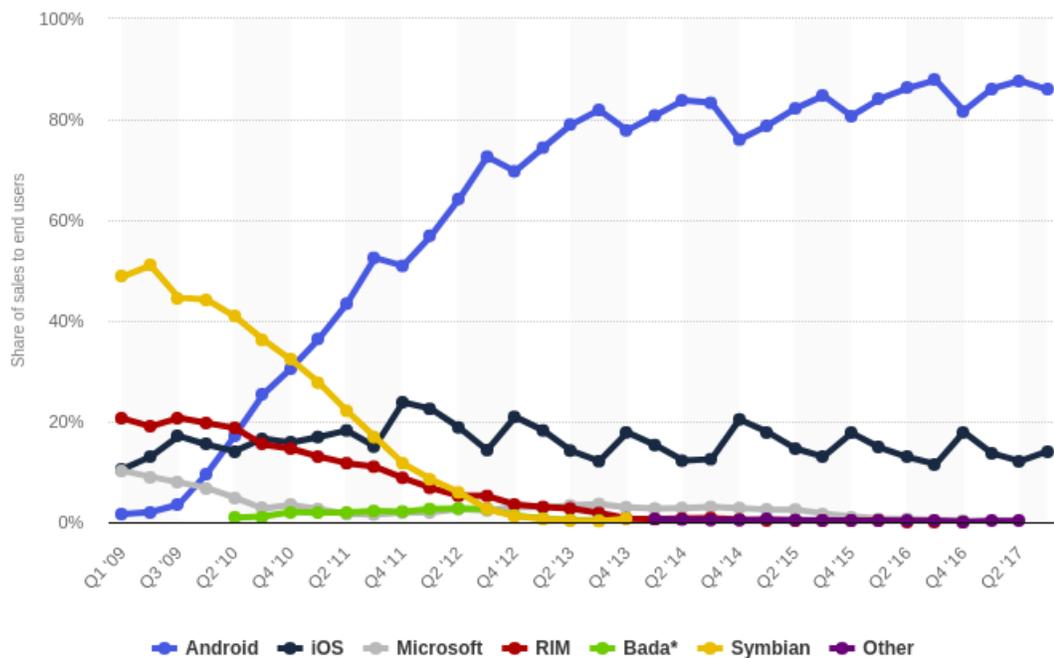
sistemas emergentes, así como lenguajes de programación disponibles, hasta el tipo de aplicaciones que se pueden desarrollar, entre otros. Fundamentos que ayudan a definir el tipo de desarrollo que se desea realizar pensando en maximizar el retorno de inversión proyectado.

2.4.1. Sistemas operativos

Cuando se pretende desarrollar una aplicación móvil se deben considerar los sistemas operativos que existen. Ya que cuantos más sistemas sean soportados mayor será el número de usuarios a los que se podrá alcanzar. Sin embargo, no siempre interesa darles soporte a todos los sistemas operativos, dependiendo del negocio y sus objetivos puede que con uno o dos sistemas sean suficientes para solventar las necesidades de este.

En los últimos 10 años, han surgido varios sistemas operativos, como Android, IOS, Windows Phone OS, BlackBerry OS, Firefox OS, Symbian, entre otros. En la actualidad, los sistemas más populares son Android (específicamente Google Android, más de esto en la siguiente sección) y IOS, ocupando un tercer puesto Windows Phone OS. Google con su versión de Android posee más del 80 % del mercado mientras que Apple con IOS y en segundo lugar, apenas un 18 %, (figura 5). Por eso, cuando se desea desarrollar una aplicación, la atención se centra en estos dos principalmente.

Figura 5. **Marketshare por sistema operativo en los últimos años**



Fuente: Statista. *Global mobile OS market share in sales to end users.*

<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>. Consulta: 05 de junio 2019.

2.4.1.1. **Android**

Android es un sistema operativo desarrollado inicialmente por la compañía Android Inc. fundada en el 2003 por Rich Miner, Nick Sears, Chris White, y Andy Rubín. Los fundadores declararon que Android fue concebido para mejorar los sistemas operativos de las cámaras digitales. Sin embargo, meses después decidieron hacer de este un sistema operativo para móviles.

Para el 2005, Google adquirió Android Inc. Aun así, Rubín y otros fundadores se quedaron a cargo del desarrollo bajo las directrices de sus nuevos

propietarios. Una de estas fue basar Android en Linux, así se podría ofrecer de manera gratuita. Google consideraba que podía generar ganancias por medio de sus servicios y aplicaciones que desarrollan.

La primer versión beta para desarrolladores fue liberada a finales del 2007. Ya que Android es un SO de código abierto, esto supone que cualquier persona puede contribuir al mismo. También supone que cualquier compañía puede desarrollar su propia versión, hacerla privativa y cobrar por su sistema como mejor les convenga. Por ello, al hablar de Android, se aclara que la versión más popular es la desarrollada, mantenida y autorizada por Google.

Por esta razón, es importante definir el mercado objetivo para determinar si depender de alguna herramienta provista por Google resulta un riesgo o no. Aun así, la versión de Google sigue siendo la más utilizada en la actualidad, con un panorama complicado para que esto cambie. Así que muchas veces se opta por soportar dicha versión.

2.4.1.2. IOS

Es el sistema operativo de las terminales móviles producidas por Apple. Durante el anuncio del iPhone original, Apple afirmó que su nueva terminal móvil correría con el mismo núcleo Unix que su sistema operativo para computadoras personales Mac OS X y que utilizaría muchas de las mismas herramientas. Sin embargo, fue claro que, aunque había muchos elementos compartidos entre OS X y este nuevo sistema operativo para móviles, este nuevo producto era lo suficientemente diferente como para ameritar su propia marca. Cuando fue lanzado el primer iPhone, el sistema operativo con el que corría fue llamado iPhone OS y mantuvo ese nombre durante cuatro años, cambiándolo a iOS en el 2010 con el lanzamiento de iOS 4.

Cuando el iPhone fue introducido por primera vez, está muy por detrás de su competencia. Windows Mobile, Palm OS, Symbian e incluso BlackBerry eran sistemas ya bien establecidos en el 2007, con una amplia gama de características. En comparación, el iPhone no soportaba redes 3G, no tenía multitareas, no tenía opciones de copiar y pegar, no se podía adjuntar cualquier archivo a un correo electrónico, no tenía soporte para MMS, no se podía personalizar la pantalla de inicio. Pero a Apple no le importaban estas características y todo el mundo lo sabía. En lugar de competir en esa área, Apple se enfocó en hacer bien la experiencia de usuario en su nuevo sistema. Le dio prioridad a la velocidad, consistencia entre aplicaciones y en hacer algunas características radicalmente mejores que lo que ya estaba disponible en 2007.

2.4.2. Diferencias entre una aplicación web y una aplicación móvil

Una aplicación móvil es aquella que se instala directamente en el dispositivo, esto supone la existencia de un binario compilado específicamente para la plataforma en cuestión. Por otro lado, una aplicación Web funciona a través de un Navegador Web en el teléfono, pero requiere una señal de celular o Wi-Fi para descargar los recursos que la conforman.

Otra diferencia que presentan las aplicaciones nativas es la utilización de los recursos del dispositivo. Una aplicación nativa puede funcionar mucho más rápido al aprovechar la potencia del procesador sin las limitaciones del navegador. Además, puede acceder a hardware específico como GPS, sensores de movimiento, cámaras, detectores de huellas dactilares, entre otros. En algunos teléfonos inteligentes, la aplicación puede controlar dispositivos y actuar como un controlador.

La aplicación Web tiene la fortuna de ser utilizada en varios dispositivos con el único requisito de ser un navegador Web y una conexión a Internet. Siempre y cuando la tecnología sea soportada por el navegador. El dilema es que, a diferencia de la aplicación nativa, requiere acceso a internet y su velocidad dependen de la calidad de la señal y ancho del ancho de banda de la conexión que posea. Otro problema es que no tendrá acceso a hardware interno como el GPS y otras funcionalidades.

De modo que todo se reduce a si desea obtener una mayor gama de herramientas, funcionalidades y poder de procesamiento o si se desea incrementar su portabilidad.

2.4.3. Tipo de aplicaciones según su desarrollo

Existen diferentes maneras para clasificar una aplicación. Podemos clasificar por interfaz, arquitectura, origen de datos, o por la naturaleza de esta. A continuación, se presentan los tipos de aplicaciones según su arquitectura.

2.4.3.1. Aplicaciones nativas

Se les conoce como aplicación nativa a todas aquella que han sido desarrolladas con el software que cada SO ofrece a los programadores. Cada plataforma ofrece un SDK (Software Development Kit) sobre el cual se codifica la aplicación para su posterior compilación. Así el SDK genera un archivo binario estructurado de manera que el SO pueda instalarlo y ejecutarlo.

Cada SDK es propio del sistema operativo, por ello, en la sección anterior se mencionaba cómo, una aplicación nativa, es más difícil de portar. Si se desea desarrollar una aplicación nativa para dos sistemas operativos, se deberá

mantener dos proyectos específicos para cada uno. Esto aumenta la complejidad, tiempo y costo de desarrollo.

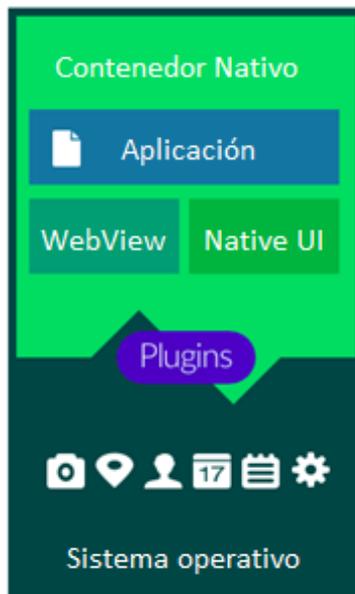
2.4.3.2. Aplicaciones híbridas

Las aplicaciones híbridas son una mezcla entre una aplicación nativa y una web, por ello su nombre. Estas son desarrolladas usando una mezcla de tecnologías Web como HTML, CSS y JavaScript. La diferencia principal es la arquitectura interna ver figura 6 que estas aplicaciones requieren para ejecutarse. Esta arquitectura les permite acceder al hardware del dispositivo, como el acelerómetro, la cámara, los contactos y más. Estas son capacidades que a menudo están restringidas al acceso desde navegadores móviles.

Para el usuario común resulta muy difícil diferenciar entre una aplicación híbrida y una nativa. En general, una aplicación híbrida, correctamente diseñada, dará la misma sensación que una aplicación nativa. Sin embargo, para obtener los mejores resultados casi siempre se deberá invertir una cantidad de tiempo considerable al diseño de esta.

En la actualidad, la mayoría de las aplicaciones móviles híbridas usan Apache Cordova, una plataforma que proporciona un conjunto de APIs para acceder al *hardware* del dispositivo a través de *plug-ins*, codificados de manera nativa. El siguiente esquema muestra los componentes que interactúan en una aplicación híbrida.

Figura 6. **Arquitectura interna de una aplicación híbrida**



Fuente: elaboración propia.

2.4.3.3. Ventajas y desventajas del desarrollo de una aplicación híbrida vs. una nativa

Como se ha mencionado, la mayor ventaja de una aplicación híbrida sobre una nativa es la portabilidad. Esta se codifica en un lenguaje en común y se compila con diferentes SDKs para que se ejecute en sus respectivas plataformas.

Partiendo de lo anterior, el desarrollo de aplicaciones híbridas resulta atractivo para las empresas desde un punto de vista de costos de mano de obra y tiempo de desarrollo, ya que se puede contratar a un solo desarrollador que conozca de tecnologías más comunes como HTML, CSS y JavaScript para generar un único *base code* capaz de correr en más de una plataforma. Y,

aunque en la práctica siempre se requerirá de conocimientos básicos de las demás plataformas, contratar a un experto ya no es necesario.

Sin embargo, por otro lado, una desventaja que presentan es la eficiencia con la que usan los recursos del dispositivo. Debido a que el entorno móvil posee una memoria limitada y capacidad de procesamiento limitada, el propósito de los SO es coordinar por medio de varios mecanismos a todas sus aplicaciones. Sin embargo, una aplicación híbrida restringe este comportamiento haciendo trabajar de manera ineficiente al SO.

Así pues, desde el punto de vista de desarrollo y mantenimiento, una aplicación híbrida puede resultar sustancialmente más rentable que varias aplicaciones nativas. Sin embargo, es importante determinar la clase de tareas que ésta estará realizando para así determinar la demanda de recursos que se le harán al SO. Si la velocidad, eficiencia e innovación son claves para el éxito de la aplicación puede que una aplicación híbrida no sea la mejor opción.

2.4.4. Progressive web apps

El término Progressive Web App (PWA) fue usado por primera vez en el 2015 para describir aplicaciones web que hacían uso de las últimas tecnologías de los navegadores más modernos como lo son los *service workers* y web app manifests. Esta es una tecnología emergente, que busca reducir la brecha entre aplicaciones Web y aplicaciones nativas.

Es común para las aplicaciones nativas hacer cosas como enviar *push notifications*, trabajar sin conexión, cargar en la pantalla de inicio, etc. Las aplicaciones web a las que se accede desde un navegador móvil, en comparación, no pueden seguir con el mismo comportamiento. Las PWAs lo

solucionan con nuevas APIs, nuevos conceptos de diseño y varios conceptos nuevos.

Las PWAs no han de ser confundidas con una aplicación híbrida. Estas últimas, se ejecutan sin inhibiciones en el entorno de aplicaciones nativas, con muchas de las características que se esperan de cualquier aplicación nativa. Pero si a esta se le extrae el contenido web y se carga en un navegador móvil, la aplicación queda limitada por las características del navegador (por una variedad de buenas razones de seguridad y estandarización API).

Las PWA son capaces de funcionar fuera de línea y actualizar su contenido sin intervención del usuario. También pueden recibir *push notifications* y, especialmente, el hecho de que son instalables en el dispositivo del usuario hace de estas aplicaciones una opción atractiva para muchos desarrollos dependiendo de los requerimientos y los alcances que se deseen obtener.

En general, las aplicaciones web progresivas describen una colección de tecnologías, conceptos de diseño y API web que funcionan en conjunto para proporcionar una experiencia similar a la aplicación nativa.

3. BENCH MARK

Como se ha evidenciado en la sección anterior, los problemas que presenta una PYME dedicada a la venta de alimentos y bebidas no son propiamente de este tipo de negocios. Incluso, las grandes cadenas de alimentos se enfrentan con retos similares. Sin embargo, una PYME invierte menos recursos en la innovación por falta de conocimiento o capital. Esta representa una gran debilidad si se compara con una marca ya establecida.

A continuación, se presenta un estudio de mercado de las soluciones disponibles. Para cada solución se pretende definir: su costo, complejidad (curva de aprendizaje), control, *framework* y sus funcionalidades. La siguiente tabla enmarca de manera más específica lo que se pretende descubrir para cada una de las soluciones.

Tabla II. **Criterios de evaluación del benchmark**

Área	Criterios
Costo	<p>Se refiere a los costos en los que una institución o el usuario final incurrirán para hacer uso del software.</p> <p>Este criterio evalúa características como:</p> <ul style="list-style-type: none"> • Requerimientos del sistema • Infraestructura necesaria • Licencias • Instalación
Complejidad	<p>Evalúa lo compleja que puede llegar a resultar la curva de aprendizaje de la herramienta provista por el software. En general se evalúa la usabilidad.</p> <p>Existen características que indirectamente influyen en esta área como:</p> <ul style="list-style-type: none"> • Disponibilidad de soporte técnico • Herramientas externas
Control	<p>La seguridad, personalización y privacidad son los puntos por evaluar para medir el nivel de control sobre la solución.</p>
Claridad	<p>Acá se evalúa la interfaz del usuario y accesibilidad (resolución, sonido, tamaño, colores, legibilidad, entre otros).</p>
Framework	<p>Se refiere al marco de trabajo en relación con la Interoperabilidad, escalabilidad, plataformas, integración.</p>
Funcionalidades	<p>Evalúa la funcionales en general.</p>

Fuente: elaboración propia.

3.1. Loyverse

Loyverse es un punto de servicio que se constituye a partir de varias aplicaciones. Cuenta con un punto de venta Loyverse TPV para móviles y tabletas, con un BackOffice administrativo. También cuenta con una aplicación dedicada a visualizar los indicadores del negocio, otra aplicación para mostrar las órdenes en la cocina y, finalmente, una aplicación para que el cliente pueda visualizar su orden y solicitar el envío del recibo a su correo. En total, el punto de servicio lo integran cuatro aplicaciones completamente gratis y un sitio administrativo.

Loyverse cuenta con funcionalidades adicionales que se habilitan por medio de una suscripción de \$ 25 USD. Funcionalidades como la administración de personal e inventario son algunas de las que requieren dicha suscripción.

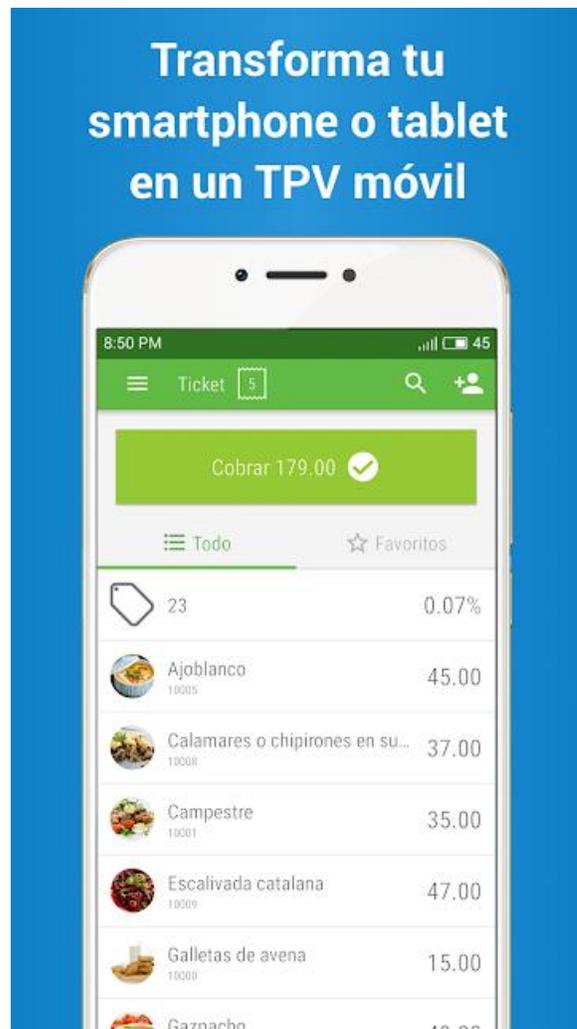
Montar un punto de venta con Loyverse es relativamente rápido y sencillo. La parte más compleja es la configuración de la empresa desde el sitio administrativo. La versión gratuita es bastante completa con el único inconveniente que no se podrá asignar un usuario a los diferentes cajeros.

El mal diseño de la aplicación para dispositivos más pequeños propicia que el flujo de captura de un pedido y el cobro de este resulte confuso y difícil de entender. Un mejor diseño y una mejor usabilidad genera en los empleados una reducción en la curva de aprendizaje y menor resistencia al cambio.

Loyvere se destaca porque brinda un abanico bastante completo de funcionalidades para que el propietario pueda controlar todos los aspectos de su negocio. Desde reporte de ventas, configuración de artículos para la venta y noventa, registro de inicio y fin de turnos por empleado, hasta la administración

de clientes. Sin duda alguna, el punto de servicio de Loyverse posee las funcionalidades básicas y las excede.

Figura 7. Loyverse IU



Fuente: Loyverse. *Free point of sale and Inventory Management Software.*
<https://loyverse.com/>. Consulta: 20 de junio 2019.

3.2. Kyte

Kyte es un punto de venta que se compone de una única aplicación. Está disponible para dispositivos con sistema Android o IOS. A pesar de no tener un sitio administrativo como la solución anterior, cuenta con muchas funcionalidades y herramientas similares. Entre las herramientas en común están el registro de clientes, envío de recibos por correo, administración de usuario y reportes estadísticos de las ventas.

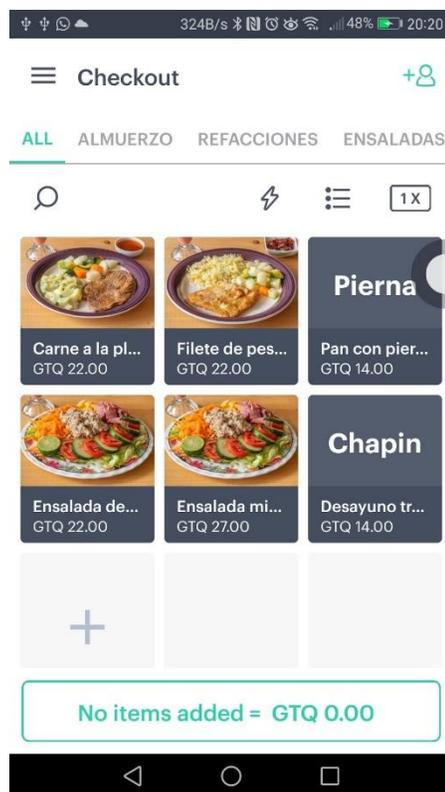
A diferencia de la solución anterior, todas las funcionalidades de este punto de venta son completamente gratuitas. La instalación requiere únicamente un teléfono inteligente o tableta con sistema Android o IOS. Es importante recalcar que puede funcionar completamente fuera de línea, por lo que el negocio puede reducir gastos al no pagar paquetes de internet, conscientes de que funcionalidades como el envío de recibos por correo ya no estarán disponibles.

Kyte se puede integrar con diferentes dispositivos para extender su funcionalidad. Se puede integrar con impresoras térmicas para la impresión de las comandas y recibos. También cuenta con la opción de integrar un lector de tarjetas de crédito y débito, *Sumup*. Con este lector el negocio cuenta con la opción de aceptar métodos de pago con tarjetas Visa, Master Card y American Express, entre otras, con una comisión del 1,5 %. Sin embargo, dicho lector tiene un costo de 39 €* más envío lo cual podría ser un poco elevado para ciertos negocios.

La aplicación se destaca por su diseño moderno e intuitivo. Posee un flujo bastante claro que permite su configuración en minutos. El diseño hace que la curva de aprendizaje sea mínima, acelerando el proceso de adaptación de los empleados.

Una funcionalidad peculiar de Kyte es la venta rápida de productos no registrados. Es decir que Kyte contempla la posibilidad de vender un producto o servicio que no se ofrece en el catálogo. Para una cafetería o restaurante esto es sumamente valioso debido a que muchas veces se crean platillos especiales para cumplir con los requerimientos de los clientes más exigentes.

Figura 8. **Kyte IU**



Fuente elaboración propia.

Sin embargo, Kyte no ofrece la opción de componer productos con aditamentos. Es decir, en el caso de una cafetería donde, generalmente, el cliente puede crear su platillo escogiendo entre varios acompañamientos, el vendedor

tendría que recurrir a anotaciones por cada pedido. Esto haría más lenta la toma de pedidos y podría generar muchos errores dentro del proceso.

4. DISEÑO DE LA APLICACIÓN

El diseño de la aplicación se ha de realizar meticulosamente para poder satisfacer tanto al usuario como al negocio. Para diseñar la aplicación, se estudiaron varios restaurantes tipo cafetería dentro de la ciudad de Guatemala. Así se establecieron las funcionalidades requeridas para completar un proceso de captura, preparación, entrega y cobro en cada una.

Uno de los principales retos del diseño surge de que las PYMES son empresas cuyos procesos, con frecuencia, aún no han madurado ni se han establecido formalmente. Para el sector de restaurantes esto implica que el proceso de captura, preparación, entrega y cobro de la orden varíe frecuentemente en el orden que se ejecuta cada tarea. Por ello, la aplicación se diseñó para que el usuario navegue de diferentes maneras para completar cada una de estas etapas según le convenga al negocio.

Por otra parte, los platillos que estos restaurantes suelen ofrecer son más flexibles de lo que acostumbran las grandes cadenas; es decir, un platillo puede suele modificarse al gusto del cliente quitando, agregando o reemplazando ingredientes. Esto es importante para el diseño de la solución ya que debe ser tan dinámica como para ofrecer platillos personalizados.

A continuación, se describe con detalle el conjunto de funcionalidades, el diseño y la arquitectura del sistema.

4.1. Prototipo

A continuación, se describe el prototipo de la aplicación.

4.1.1. Autenticación

La pantalla de autenticación figura número 9 es la primera que se muestra al usuario cuando no ha iniciado sesión. En ella, se presenta el logo de la empresa, el formulario donde deberá ingresarse el usuario y contraseña, un botón por el cual pueden hacer visible u ocultar la contraseña y el botón que envía los datos.

El formulario valida que ambos campos sean ingresados para habilitar su envío. Al enviarse, el servidor autentica al usuario proveyendo a este un token con el cual se ha de identificar posteriormente ver figura 9.

Figura 9. **Pantalla de autenticación**



Fuente: elaboración propia.

4.1.2. Órdenes

Las órdenes representan el objeto de negocio principal del sistema. Por ello, para su manejo se requieren acciones que van desde la captura hasta la entrega y cobro. La orden pasa por un proceso donde intervienen varios actores. Por esta razón, este proceso se beneficia si se le provee a cada interviniente de los artefactos necesarios que faciliten la comunicación y así la coordinación del personal involucrado.

Para esto. cada parte del sistema se ha diseñado minuciosamente de manera que el proceso de captura, preparación, entrega y cobro dentro de la aplicación sea intuitivo, claro y rápido. Cada acción realizada sobre una orden ha sido claramente identificada y posicionada de manera que el personal interesado posea un fácil acceso a estas dentro de su área de trabajo.

4.1.2.1. Lista de órdenes

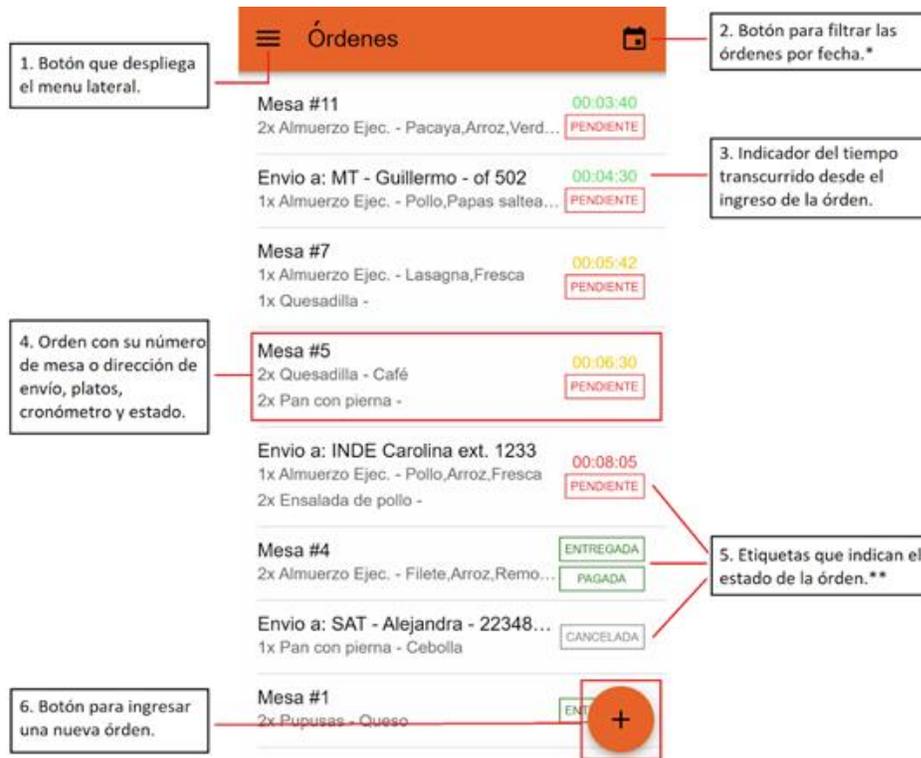
En esta pantalla se enlistan las órdenes desde la más reciente hasta la más antigua. Por cada orden se muestra el destino (número de mesa o dirección de envío), una lista de platos solicitados, el estado de la orden y el tiempo transcurrido desde su captura.

El destino de la orden puede ser un número de mesa o una dirección de envío, si el restaurante posee dicho servicio.

El estado de la orden se refiere a su estado de preparación (pendiente, entregada o cancelada). Además, junto con este también se muestra el estado de pagada, cuando está ya ha sido cobrada. Tanto la entrega, como el cobro son independientes para que el negocio pueda definir su propio modelo de entrega-cobro o viceversa.

Además, como se observa en la figura número 10, cuando la orden aún está pendiente de entrega se muestra un cronómetro que indica el tiempo que ha transcurrido desde su ingreso al sistema. Este cronómetro se pinta en forma de semáforo, el cual cambia de verde a amarillo y a rojo conforme el tiempo de espera incrementa. El negocio puede definir los tiempos que considera como bueno (verde), aceptable (amarillo) y malo (rojo) desde la pantalla de configuración.

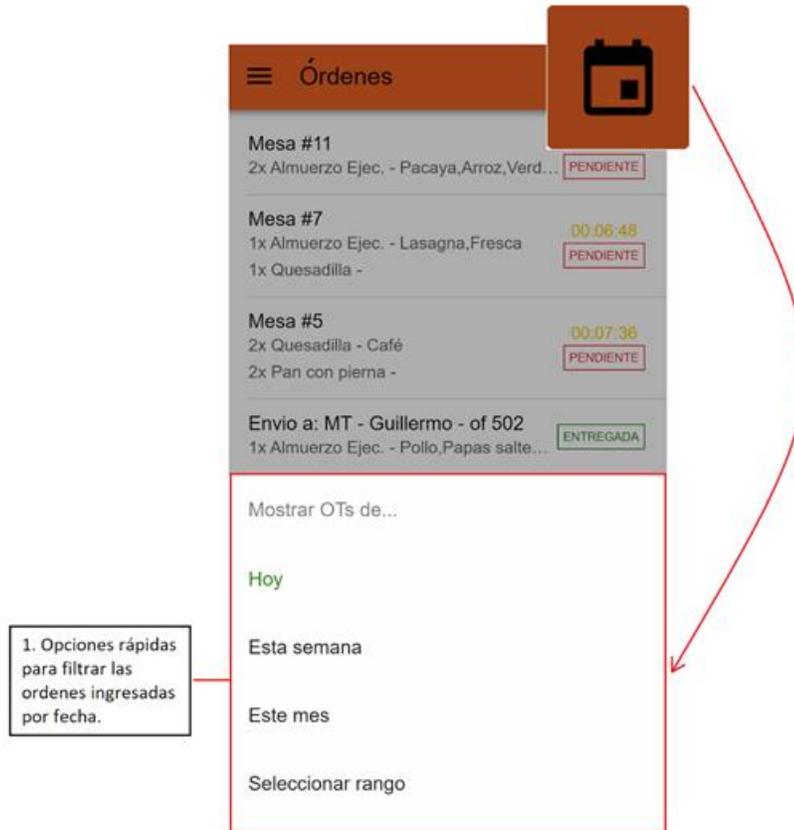
Figura 10. Pantalla de órdenes



Fuente: elaboración propia.

El botón que se muestra en la figura número 11 despliega un menú con opciones para filtrar las órdenes del día, de la semana, del mes o dentro de un rango de fechas. Por defecto el sistema muestra las órdenes del día.

Figura 11. Opciones de visualización de órdenes



Fuente: elaboración propia.

En esta lista, por cada orden, se encuentran las opciones para eliminar (cancelar), editar y marcar como pagada. Estas son visibles al deslizar con el dedo hacia un costado u otro la orden deseada, como se muestra en la figura número 12.

Se pretende que la cocina y la caja tengan acceso la misma lista de órdenes. Sin embargo, únicamente la caja o los meseros deberían estar interesados en estas opciones. Por eso, su acceso se ha diseñado para evitar que desde la cocina se ejecuten acciones que, por su naturaleza, no corresponden a esta área.

Figura 12. Opciones por orden



Fuente: elaboración propia.

4.1.2.2. Creación y edición de una orden

La creación y edición de una orden comparten el mismo formulario. La diferencia está en su encabezado. Para cuando se está editando una orden, aparece el identificador único de la misma acompañado del *destino* al que va dirigida.

El formulario consta de diferentes partes, las cuales se describen a continuación siguiendo la numeración de la figura número 13.

- Botón para cerrar: al presionarlo cierra la pantalla sin guardar ningún cambio.
- Botón para guardar: este botón se habilita hasta que la orden posee un destino válido y al menos un plato a entregar.
- Interruptor de tipo de pedido: este botón indica el tipo de pedido. Un pedido puede ser para consumir dentro de local o para llevar a domicilio. Esta funcionalidad está disponible por defecto, sin embargo, se puede desactivar en la pantalla de configuración por si no se cuenta con servicio a domicilio.

- Cuando el pedido es para llevar a domicilio, el precio adicional aparece ver figura número 14. Este precio también es configurable.
- Número de mesa: se solicita el número de mesa únicamente cuando el pedido es para comer dentro del negocio.
- Listado de platos: para cada plato se indica la cantidad de platos, el nombre del plato, y la preparación deseada, ver sección de *Plato*, junto con apuntes adicionales que el mesero pudiera indicar. Así mismo se muestra el total del plato.
- Notas: apuntes a nivel de orden. Por ejemplo, cambio que se debe dar al cliente.
- Botón flotante: botón para agregar un nuevo plato. Este botón lleva al menú de platos, desde el cual se puede seleccionar un plato para agregar a la orden.
- Total: muestra el total de la orden sumando el precio de cada plato más el costo de envío si hubiese.

Figura 13. **Formulario de orden**

The image shows a mobile application interface for creating an order. It features an orange header with a menu icon, the text 'Ordenes', a save icon, and a close icon. Below the header, the order details are shown: 'Orden AvJ57T - mesa 11'. There is a toggle switch for 'Pedido a domicilio' and a dropdown menu for 'Mesa #' set to '11'. A list of ordered items is displayed under the heading 'Platos', including 'Almuerzo Ejec.', 'Pacaya', '2x Arroz', 'Verduras (GTQ2.00)', and 'Con la salsa por aparte', with a total price of 'GTQ48.00'. Below the items is a text field for 'Notas de la orden'. At the bottom, there is a large orange '+' button and a grey bar showing the total 'Total GTQ48.00'. Eight numbered callout boxes point to various elements: 1. Close button, 2. Save button, 3. Delivery toggle, 4. Table dropdown, 5. Item list, 6. Notes field, 7. Add button, 8. Total bar.

1. Botón para cerrar orden.

2. Botón para guardar orden.

3. Campo que indica el tipo de orden.

4. Campo para indicar la mesa a la que se ha de llevar la orden.*

5. Listado de platos ordenados donde se indica la cantidad, preparación, notas y precio total.

6. Notas generales de la orden.

7. Botón para agregar más platos a la orden.

8. Total de orden.

Fuente: elaboración propia.

Figura 14. **Campo emergente para pedidos a domicilio**

The image shows a mobile application interface for a delivery order. It features a toggle switch for 'Pedido a domicilio' which is turned on. Below it, there is a text field for 'Costo de envío (GTQ)' with the value '1'.

Pedido a domicilio

Costo de envío (GTQ) 1

Fuente: elaboración propia.

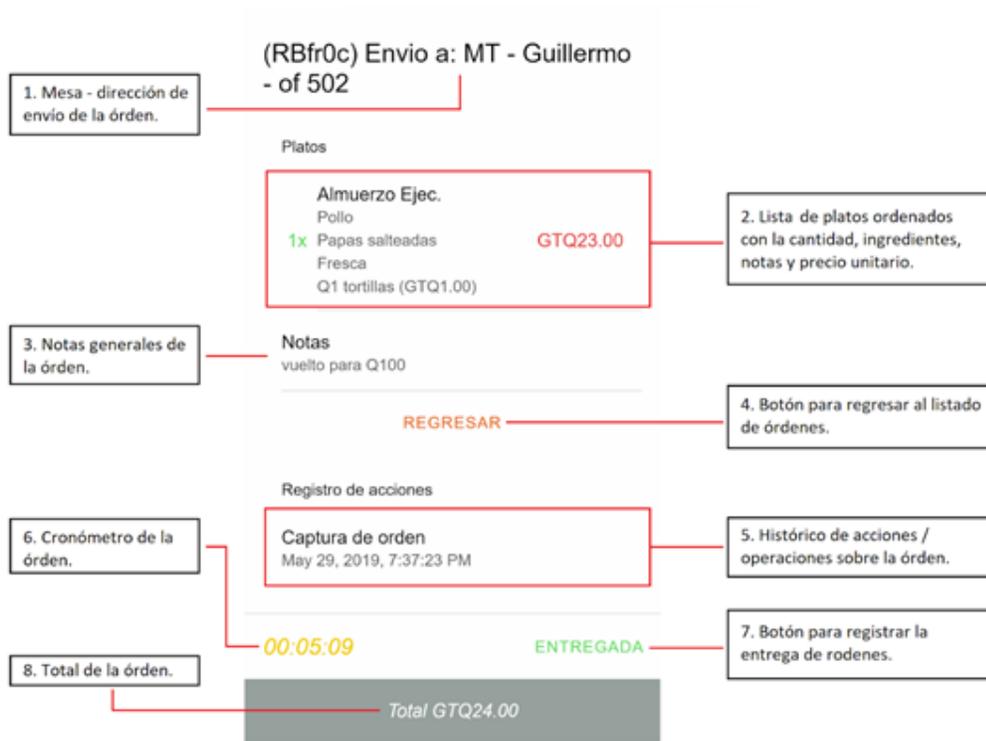
4.1.2.3. Despliegue de orden

Esta pantalla muestra la orden completa. Desde su destino, lista de platos, notas de preparación, hasta el tiempo de espera, la opción de marcar como entregada la orden y su total.

Esta pantalla está diseñada para que desde la cocina la información sea legible y clara. Por eso, se hace énfasis en el tiempo transcurrido desde su ingreso. Así mismo, se brinda la opción de marcar como entregada la orden, pues se despacha desde la cocina.

Como información adicional se muestra un registro de acciones tomadas sobre la orden. Esto puede servir para solventar confusiones que surgen en el proceso de elaboración o para consulta general a posteriori.

Figura 15. Detalle de orden



Fuente: elaboración propia.

4.1.3. Menú de platillos

En esta pantalla se enlistan los platos que el negocio ofrece. Para cada uno se muestra su nombre y una breve descripción. Aquí se pueden agregar nuevos platos presionando el botón flotante, inciso 4 según la figura número 16.

Figura 16. Menú



Fuente: elaboración propia.

4.1.4. Plato

Esta pantalla se muestra cuando el negocio desea revisar un plato para editarlo o eliminarlo y cuando un mesero está ingresando una orden.

Como se muestra en la figura número 17, se muestra el nombre del plato, su descripción, las opciones de preparación configuradas, el precio del plato (el cual puede variar según la preparación seleccionada) y un botón para agregar a la orden. Así mismo, se muestran las opciones para editar y eliminar el plato.

Cuando se está ingresando una orden, el mesero ha de seleccionar las opciones de preparación del plato. Posteriormente, se verá que estas opciones de preparación pueden configurarse para indicar acompañamientos, ingredientes extras o, incluso, ingredientes que se deben quitar.

Es importante que el negocio comprenda las necesidades del cliente para brindarle al mesero las opciones de preparación comúnmente solicitadas. Así, este puede ingresar el platillo sin necesidad de escribir, solamente debe seleccionar las opciones disponibles. Con ello se incrementa la velocidad de atención y la claridad con la que se pasa la orden a la cocina.

Figura 17. **Formulario para agregar un plato a la orden**

The screenshot shows a mobile application interface for adding a dish to an order. The form is titled 'Almuerzo Ejec.' and includes the following elements:

- Header:** A blue bar with a back arrow, a menu icon, the text 'Menú', an edit icon, a trash icon, and a close 'X' icon.
- 1. Descripción del plato:** 'Almuerzo del día acompañado de dos guarniciones, sopa, tortillas y refresco natural.'
- 2. Opciones de preparación:** A section titled 'Plato Requerido' with four options: Pollo, Filete, Pacaya, and Lasagna (priced at GTQ3.00).
- 3. Precio del plato.*:** GTQ22.00
- Guarnición #1 Requerido:** A section with two options: Arroz and Papas salteadas.
- 4. Botón para cerrar la pantalla:** The close 'X' icon in the header.
- 5. Botón para eliminar la orden.**:** The trash icon in the header.
- 6. Botón para editar el plato:** The edit icon in the header.
- 7. Indicador si es necesario escoger una opción como mínimo:** A red asterisk next to the 'Plato Requerido' title.
- 8. Indicador del precio adicional de la opción:** The 'GTQ3.00' price next to the 'Lasagna' option.
- 9. Botón para agregar la orden al carrito (i.e. a la orden):** The orange 'AGREGAR' button with a shopping cart icon.

Fuente: elaboración propia.

* Precio del plato: inicialmente se muestra el precio base, sin embargo, este precio puede variar si se escoge alguna preparación con costo adicional ver inciso 8 de la figura número 17.

** Para eliminar un plato se ha de solicitar confirmación, como se muestra posteriormente.

4.1.4.1. Formulario para la creación y edición de un plato

El formulario para la creación y edición de un plato incluye:

- Un nombre
- Una descripción
- Un precio base
- Una lista de opciones de preparación

Para guardar un plato, solo se necesitan los primeros tres campos. Sin embargo, como se mencionó, se recomienda configurar tantas opciones de preparación como sea necesario para que el mesero escriba la menor cantidad de notas.

En la figura número 18 se observa cómo, cada opción, posee un botón al costado, el cual, al presionar y dejar sostenido, permite mover el ítem para modificar el orden en que estas opciones se presentan.

Figura 18. Formulario de un plato

1. Campos generales del plato: nombre, descripción y precio base.

2. Opción de preparación con las diferentes elecciones para cada una.

3. Botón para cerrar la pantalla.

4. Botón para guardar el plato.

5. Botón para agregar una opción para el plato.

6. Botones para reordenar las opciones de preparación.

Fuente: elaboración propia.

A continuación, se presenta el formulario de cada opción de preparación.

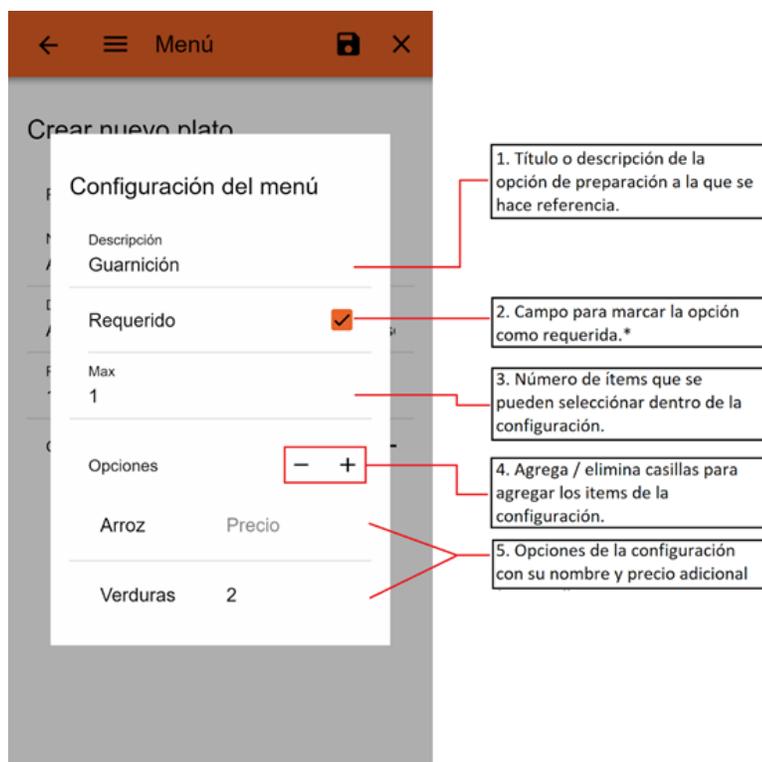
Estas opciones están conformadas por una descripción. Esta descripción puede ser un título, una pregunta o una sugerencia. El texto que se decida introducir ha de ser coherente con la intención de los ítems que se agreguen.

Además de la descripción, se ha de indicar si es requerido o no. Una opción de preparación requerida, le indica al mesero que ha de seleccionar, por lo menos, un elemento de la lista disponible. Así mismo, se puede configurar la

máxima cantidad de elementos que se seleccionarán. Una cantidad negativa indica que se puede seleccionar N cantidad de elementos.

Finalmente, cada elemento contendrá una descripción y, opcionalmente, un precio. Si el elemento no tiene costo adicional se puede dejar en blanco.

Figura 19. **Formulario para agregar una opción de preparación**



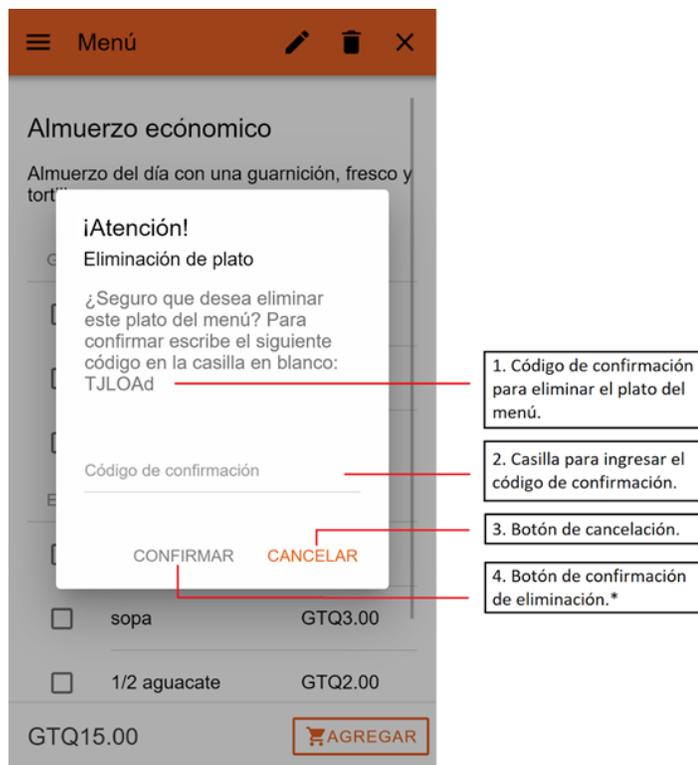
Fuente: elaboración propia.

4.1.4.2. Eliminación de un plato

Como se mencionó, al presionar el botón del inciso 5 de la figura número 17, se despliega una alerta solicitando la confirmación. Esta acción no se puede revertir y por eso, se debe estar seguro y consciente de lo que se hace.

Para asegurar que el usuario está completamente anuente a la acción que está realizando y que ya ha leído la alerta, se le solicita que ingrese el código de confirmación. Cuando el código ingresado coincide con el solicitado, se habilita el botón de confirmación. Al presionar este último ejecuta la acción de eliminación.

Figura 20. **Confirmación para la eliminación de un plato**



Fuente: elaboración propia.

4.1.5. Factura electrónica en línea (FEL)

Tras marcar una orden como cobrada, el sistema da la opción para facturarla. Al ingresar a esta opción, se despliega una pantalla donde se muestra un formulario que solicita los datos del cliente ver figura número 21 junto con el detalle de la orden y su total.

Figura 21. Datos del contribuyente

The screenshot shows a mobile application interface for generating an electronic invoice. The header is an orange bar with a menu icon, a back arrow, and the text 'Facturas'. Below the header, the form is organized into several sections:

- Order Information:** A field labeled 'Facturar orden FgQxe3' is linked to the annotation '1. Orden a facturar.'
- Receiver Contact:** A field labeled 'Correo electrónico' is linked to '2. Correo del receptor.'
- Receiver Identification:** A field labeled 'NIT' is linked to '3. Nit del receptor o CF.'
- Receiver Name:** A field labeled 'Nombre' is linked to '4. Nombre del receptor.'
- Receiver Address:** A field labeled 'Dirección' is linked to '5. Dirección del receptor.'
- Order Items:** A section titled 'Platos' contains one item: 'Almuerzo economico' with a quantity of '1x', 'Arroz tortillas (GTQ1.00)', and a total price of 'GTQ16.00'. This section is linked to '6. Platos de la orden a facturar.'
- Invoice Generation:** A circular orange button with a document icon is linked to '7. Botón para generar la factura electrónica.'
- Total:** A grey bar at the bottom shows 'Total GTQ16.00', linked to '8. Total de la factura.'

Fuente: elaboración propia.

Una vez que el cajero completa el formulario y oprima el botón de facturación, el sistema genera un Documento Electrónico o DTE tipo factura en formato XML para que un agente certificador lo valide y firme. Luego, este DTE

firmado se envía para su certificación. El agente certificador lo envía a los servicios correspondientes de la Superintendencia de Administración Tributaria donde nuevamente es validado y autorizado. Si la factura cumple con las normas establecidas, el agente certificador envía un Identificador Único Universal UUID (Universally Unique Identifier, siglas del nombre en inglés).

El UUID no solo da validez a un DTE, también para un DTE tipo factura representa la serie y el número de esta. La serie se forma con los primeros 8 dígitos hexadecimales del UUID, de izquierda a derecha. El número de factura se forma con el equivalente en números decimales de los dígitos hexadecimales del UUID, a partir de la posición 9 hasta la posición (sin incluir guiones).

Dentro de la aplicación el UUID generado se muestra en la pantalla de órdenes dentro de cada ítem como se observa a continuación.

Figura 22. **Orden ya facturada**



Fuente: elaboración propia.

4.1.6. Parametrización del sistema

El sistema cuenta con varias características que se pueden configurar para adaptarse de mejor manera al modelo del negocio. Por ello, cuenta con una pantalla de configuración donde se encuentran parametrizados:

- El número de mesas dentro del negocio.
- Si posee o no servicio a domicilio y el costo que este posee.
- Los datos de facturación del negocio. Estos datos se enviarán en el DTE de la factura electrónica según lo requiere la SAT.

Figura 23. Pantalla de configuración del sistema



Fuente: elaboración propia.

Al seleccionar los datos de facturación, se despliega un *modal* adicional pues el formulario se compone de varios campos ver figura número 24.

Figura 24. Configuración de los datos del emisor

Datos de facturación - Emisor ✕

Configura los datos solicitados para poder emitir facturas digitales. Asegúrate de tener los datos correctos para evitar inconvenientes con la SAT.

1. Datos del emisor requeridos por la SAT.

Régimen	General ▾
NIT	1000000000K
Nombre o razón social	DEMO, S.A.
Nombre comercial	DEMO, S.A.
Código establecimiento (otorgado por I...	1
Correo	conta@gmail.com
Dirección	Avenida reforma 3-48 Zona 9 Edificio Ane
Código postal	01009
Municipio	

2. Botón para cerrar el formulario.

3. Botón para guardar los datos del formulario

Fuente: elaboración propia.

4.2. Diseño intuitivo y usabilidad

El diseño de la aplicación se basó en los conceptos y estrategias de *Material Design*. Este último es una guía propuesta por Google que recopila las mejores prácticas para la elaboración de interfaces de usuario. Siguiendo los lineamientos descritos en esta guía es posible elaborar un producto intuitivo, agradable y consistente con varios sistemas ya existentes.

En general se busca que todo el sistema sea consistente con una línea de diseño claro. Para esto se ha definido una paleta de colores, se ha escogido una tipografía e iconografía de acuerdo con el producto y se ha establecido un

sistema claro de navegación. Esto se acompaña de diferentes animaciones para las transiciones y acciones para darle continuidad a cada elemento dentro del sistema.

Además, es importante que el sistema pueda transmitir lo que está sucediendo en cada pantalla. Es decir, si el sistema está cargando información, este ha de notificar al usuario de una u otra manera. Por otra parte, si el sistema espera alguna acción por parte del usuario, este debe hacérselo saber. Así mismo, si se produce algún error por el cual el sistema no puede completar alguna acción, este debe transmitirse al usuario de una manera clara. Con esto se logra la confianza del usuario para que siga usando nuestro producto.

4.2.1. Paleta de colores

La paleta de colores consta de 9 colores: un color primario, secundario, terciario, de éxito, de advertencia, de peligro o alerta, un color oscuro, uno medio y uno claro. No es necesario usar todos los colores, sin embargo, se aconseja, por lo menos, definir el primario, secundario, de éxito de alerta y de peligro. Adicionalmente para cada color se definió un color de contraste, una sombra y un tinte.

Figura 25. Ejemplo de la paleta de colores



Fuente: elaboración propia.

Los colores empresariales son el primario, secundario y terciario. Si solo se poseen dos colores empresariales solo se definen el primario y secundario. El color de advertencia suele ser amarillo porque se usa de esta manera en diferentes industrias y, por lo tanto, el usuario lo reconocerá así naturalmente. De la misma forma, el color de peligro o alerta suele ser el color rojo ya que se suele asociar con la señal de ALTO y por ello, causa una mayor impresión, haciendo que el usuario se detenga a leer brevemente.

En la pantalla de captura y edición de la orden figura número 25, se ejemplifica el uso de los colores primarios y secundarios. Como se observa, los

elementos principales llevan el color primario, otros elementos estrechamente asociados con el propósito principal de la pantalla se colorean con el secundario y ayudas visuales de baja relevancia con un color claro. Esto permite manejar el orden en que el usuario observa los elementos dentro de la pantalla, facilitando la comprensión de cada uno y las acciones que dentro de la misma puede ejecutar.

4.2.2. Tipografía

La intención con la tipografía en cada pantalla es definir el orden en que se atrae la atención del lector. El uso correcto del tamaño de la letra, así como su estilo permite dar un sentido de unidad al contenido que se muestra. De esta manera, se ayuda para que el usuario entienda el contenido que se le presenta.

En cada pantalla se ha procurado seguir los lineamientos de *Material Design*. La figura número 25 muestra el sistema de tamaños utilizados para cada elemento.

Figura 26. **Sistema de tamaños de fuente**

Scale Category	Typeface	Font	Size	Case	Letter spacing
H1	Roboto	Light	96	Sentence	-1.5
H2	Roboto	Light	60	Sentence	-0.5
H3	Roboto	Regular	48	Sentence	0
H4	Roboto	Regular	34	Sentence	0.25
H5	Roboto	Regular	24	Sentence	0
H6	Roboto	Medium	20	Sentence	0.15
Subtitle 1	Roboto	Regular	16	Sentence	0.15
Subtitle 2	Roboto	Medium	14	Sentence	0.1
Body 1	Roboto	Regular	16	Sentence	0.5
Body 2	Roboto	Regular	14	Sentence	0.25
BUTTON	Roboto	Medium	14	All caps	1.25
Caption	Roboto	Regular	12	Sentence	0.4
OVERLINE	Roboto	Regular	10	All caps	1.5

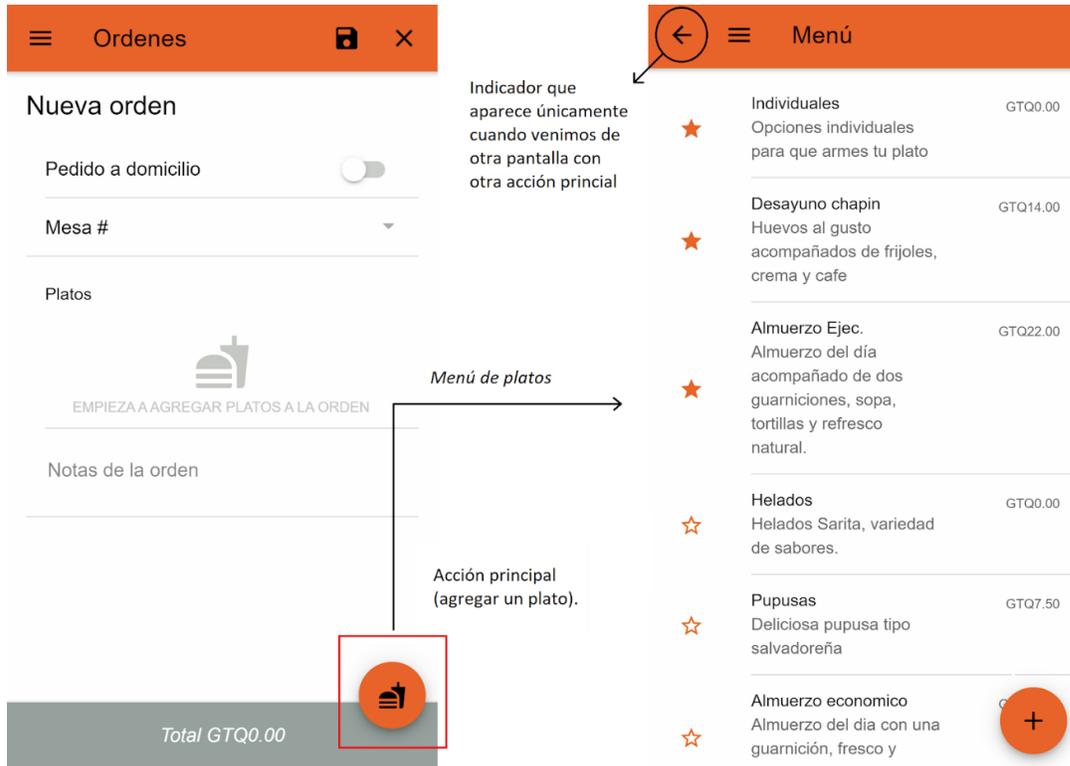
Fuente: Google LLC. Material Design. <https://material.io/design/typography>.

Consulta: 15 junio 2019.

4.2.3. Navegación

La navegación se diseñó para ser consistente y suficientemente flexible a lo largo del sistema. Es decir, por medio del flujo de las pantallas, el usuario identifica de dónde viene y para dónde debería de ir. También le permite interrumpir su flujo realizando otra operación según la necesidad que posea.

Figura 27. Navegación



Fuente: elaboración propia.

4.3. Arquitectura del software

La arquitectura de software es la estructura que se le da a un sistema. En esta se define la forma en la que se han de ensamblar los diferentes elementos para su funcionamiento. Una arquitectura se diseña con base en los objetivos, restricciones y limitantes del sistema. Un buen diseño ayuda tanto al funcionamiento como al mantenimiento del sistema.

A continuación, se presenta la arquitectura usada para la elaboración del sistema.

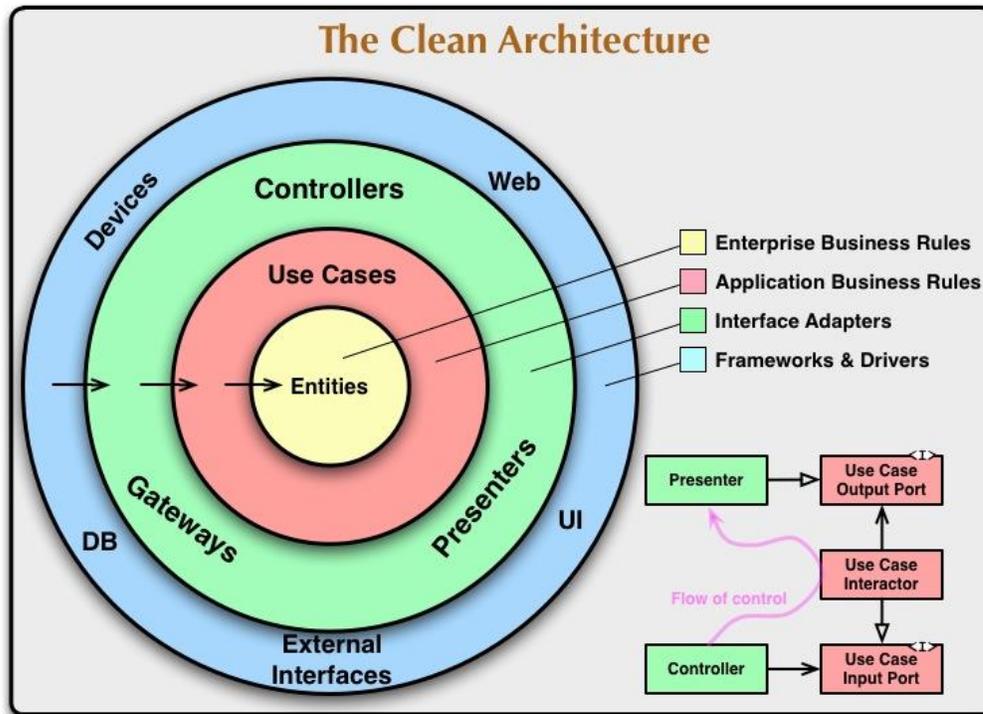
4.3.1. Clean architecture

Robert C. Martín propuso la arquitectura implementada. Se basa en la división de las responsabilidades para lo cual define “capas”. Esta arquitectura también define principios y reglas para generar sistemas que sean:

- Independientes de *frameworks*.
- Conformados con elementos que pueden ser probados (*testable*) con mayor facilidad.
- Independientes de la vista.
- Independientes de bases de datos.
- Independientes de cualquier agente externo.

En general, las reglas de negocio no deben evidenciar ninguna dependencia externa ya que son detalles de implementación que deberían ser irrelevantes al sistema. El siguiente diagrama procura demostrar a grandes rasgos lo que se trata de conseguir al seguir los lineamientos propuestos.

Figura 28. Diagrama de la arquitectura



Fuente: CECIL MARTIN, Robert . *The Clean Code Blog*.

<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>.

Consulta: 15 de junio 2019.

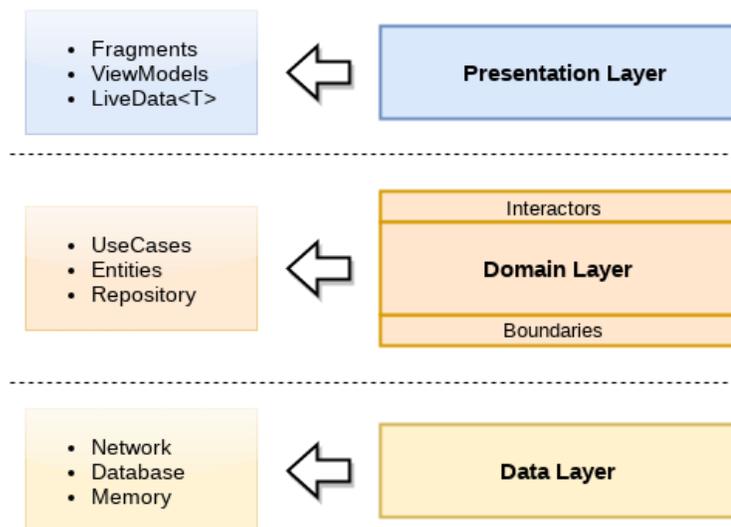
En el diagrama anterior también se define la regla de dependencia que cada capa ha de respetar. Esta indica que ninguna capa interior debe poseer una dependencia hacia el exterior y así cada capa externa dependerá de una interna apuntando con una flecha hacia el interior.

4.3.1.1. Arquitectura de 3 capas

En general, la arquitectura posee tres capas principales (presentación, dominio y datos). Sin embargo, se pueden crear tantas capas como sea necesario (siempre que tengan sentido para el proyecto/negocio).

En cada una de las capas se usa uno o varios patrones de diseño que definen la ruta para seguir construyendo las funcionalidades del sistema ver figura número 29.

Figura 29. **Capas de la arquitectura**



Fuente: CEJAS, Fernando. *Architecting Android*.

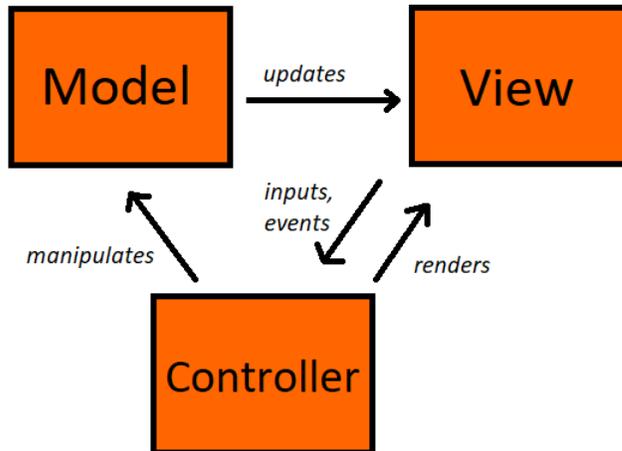
<https://fernandocejas.com/2018/05/07/architecting-android-reloaded/>.

Consulta: 15 de junio 2019.

4.3.1.1.1. **Capa de presentación: MVC**

En general, para la capa de presentación se escoge algún patrón como MVP, MVC, MVVM, entre otros. Debe ser funcional según la herramienta que se esté utilizando. En este sistema se eligió el patrón Modelo Vista Controlador ver figura 30 ya que bajo este patrón está diseñado Angular lo cual facilitará el mantenimiento de la aplicación en un futuro.

Figura 30. Patrón de diseño MVC



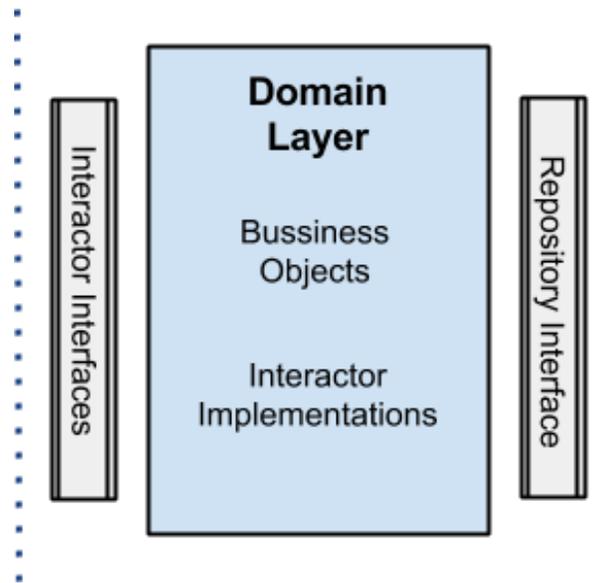
Fuente: elaboración propia.

4.3.1.1.2. Capa de dominio

En la capa de dominio se definen las reglas de negocio. Toda la lógica propia del negocio sucede aquí. En esta capa se implementan todos los casos de uso que el sistema posea.

Esta capa es la más abstracta, por lo cual carece de toda dependencia. Cualquier comunicación con agentes externos se hace por medio de interfaces definidas dentro de la misma capa.

Figura 31. **Diagrama de la capa de dominio**



Fuente: CEJAS, Fernando. *Architecting Android*.

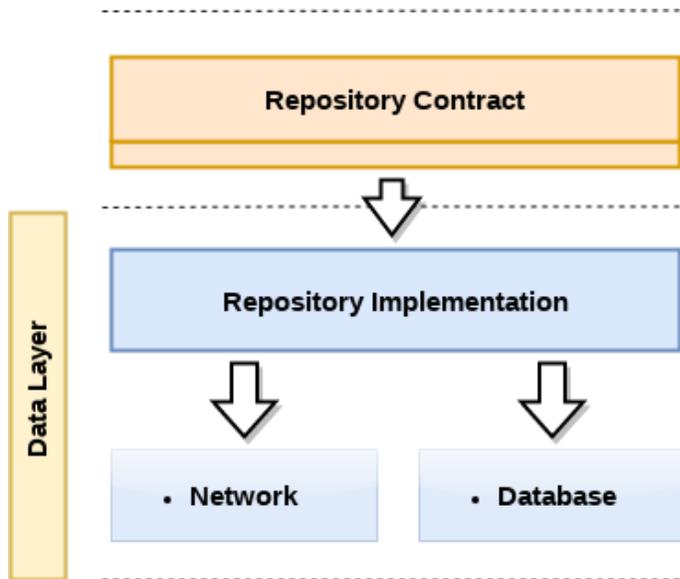
<https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/>.

Consulta: 15 de junio 2019.

4.3.1.1.3. Capa de datos: *repository pattern*

La capa de datos se ha implementado siguiendo el *Repository Pattern* para lograr una abstracción de la fuente de datos que utiliza cada caso de uso o servicio definido en la capa de dominio.

Figura 32. **Repository pattern**



Fuente: CEJAS, Fernando. *Architecting Android*.

<https://fernandocejas.com/2018/05/07/architecting-android-reloaded/>.

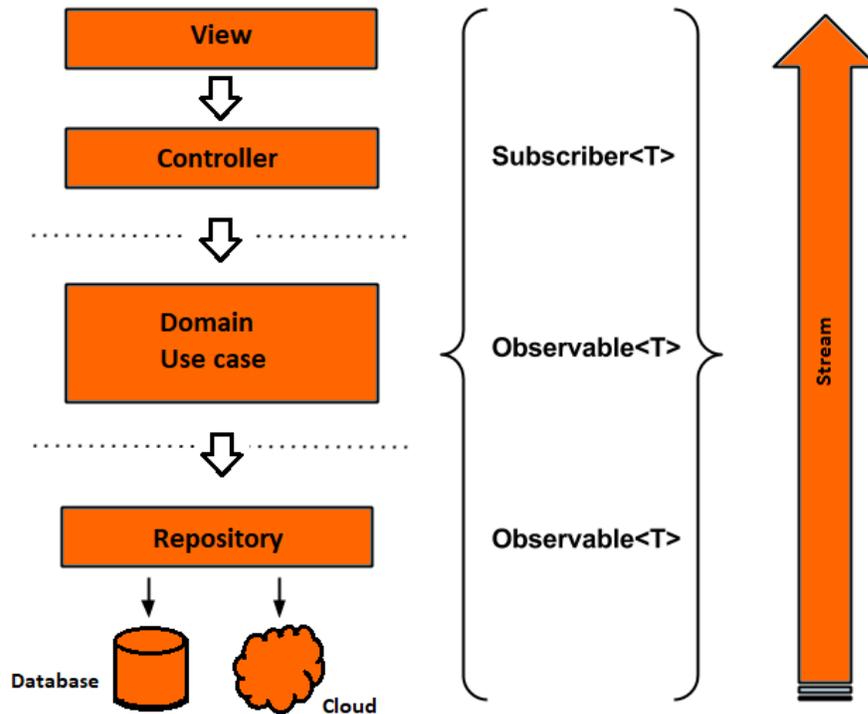
Consulta: 15 de junio 2019.

4.3.2. Flujo de datos

Para el flujo de datos se ha escogido usar RXJS aprovechando que Angular hace uso de este en gran parte de su arquitectura.

La siguiente imagen define el flujo general de los datos. La Vista por medio del Presentador / Controlador se suscribe a un caso de uso definido en la capa de dominio. Este realiza las operaciones pertinentes y envía los datos a la capa de datos por medio de un repositorio. Este repositorio se encarga, según el contexto, de enviar los datos o solicitarlos a la fuente de datos que convenga para el mismo. Por último, esta fuente de datos procesa la solicitud y devuelve una respuesta como un flujo de datos.

Figura 33. Flujo de datos



Fuente: elaboración propia.

4.4. Arquitectura del sistema

El diseño de la arquitectura se elaboró pensando en un sistema escalable, de alta disponibilidad y sencillo de mantener. Además, al ser un sistema enfocado a las PYMEs se diseñó una arquitectura que ofreciera un alto rendimiento por un costo mínimo.

Siendo un sistema que representa una gran oportunidad para el negocio, capaz de provocar resultados positivos dentro de diferentes áreas de este, se esperan nuevas características en un futuro. Así pues, la arquitectura debe ser

lo suficientemente flexible para adaptarse a estos cambios y al mismo tiempo lo relativamente sencilla de modificar y mantener.

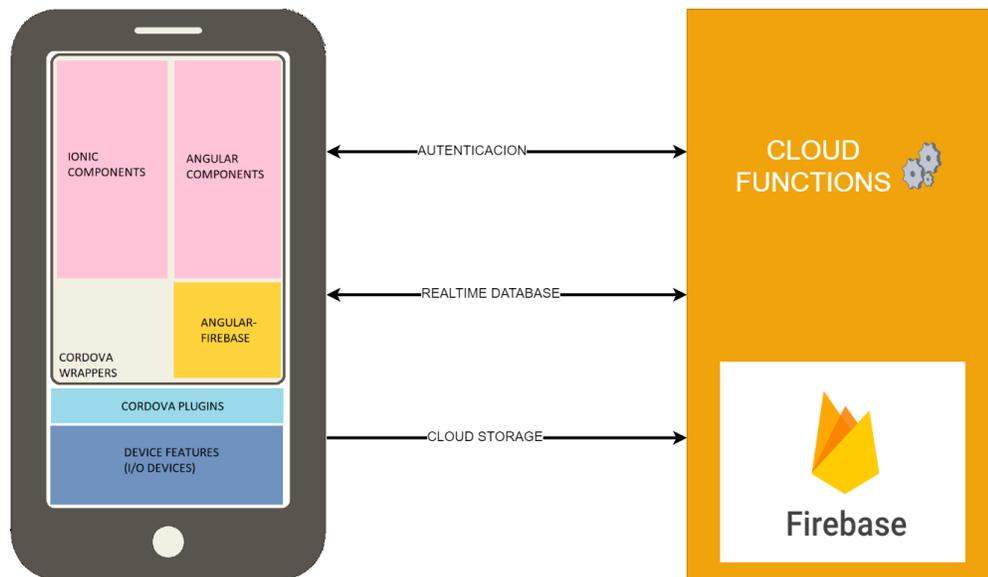
La tecnología se escogió de manera que estuviera alineada con la utilizada en el desarrollo general de la solución: tecnologías web de última generación. Así pues, el *lenguaje* a usar para comunicar las diferentes partes sería el mismo, haciendo que el mantenimiento en general del sistema sea más sencillo.

La arquitectura se divide en dos módulos principales:

4.4.1. Aplicación móvil

En la figura número 34 se muestran los componentes principales de la arquitectura que se exponen para el funcionamiento del sistema. El resto de las componentes son propios de Firebase cuyo funcionamiento se describe con mayor detalle en las secciones posteriores.

Figura 34. Módulos principales de la arquitectura



Fuente: elaboración propia.

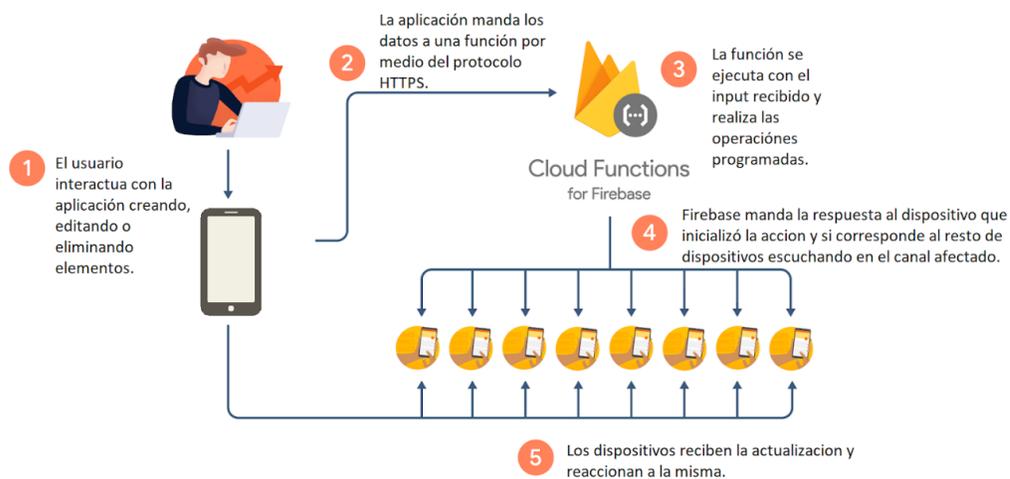
Como elemento principal del sistema se encuentra la arquitectura de la aplicación móvil. Como ya se ha establecido, esta es una aplicación híbrida, la cual hace uso de tecnologías web como lo es Ionic y Angular. Esta aplicación es encapsulada por Córdoba para que, por medio de sus interfaces y sus plugins pueda hacer uso de los dispositivos de entrada y salida como lo son la cámara, GPS, Sistema de Archivos, entre otros.

La aplicación se comunica con Firebase enviando y recibiendo información por medio de llamadas HTTP. Estas llamadas activan diferentes funciones que han sido programadas para autenticar el usuario y así como para guardar y mandar la información generada.

Firebase ofrece diferentes servicios de los cuales el sistema está utilizando la autenticación por correo, almacenamiento en la nube y una base de datos en

tiempo real. La configuración de los servicios que Firebase ofrece es relativamente sencilla. Además, el lenguaje usado es similar al que se usa cotidianamente para trabajar con tecnologías como Ionic y Angular. Esto hace facilita el mantenimiento del sistema. En la figura número 35 se ejemplifica el flujo de trabajo de ambos módulos.

Figura 35. **Flujo de comunicación entre la aplicación móvil y Firebase**



Fuente: Google LLC. *Cloud Firestore*. <https://firebase.google.com/docs/firestore>.

Consulta: 15 de junio 2019

5. DOCUMENTACIÓN

5.1. Ionic

Ionic es un *framework* de código abierto que provee un set de herramientas para desarrollar la interfaz de aplicaciones móviles y de escritorio usando tecnologías web como HTML, CSS y JavaScript.

Ionic se centra en la experiencia de usuario, entiéndase como la interacción que la interfaz de usuario presenta en la aplicación (botones, alertas, gestos, animaciones). La herramienta se caracteriza por su corta curva de aprendizaje porque se basa en tecnologías web comúnmente utilizadas. Además, con la última actualización de Ionic (v4) se puede escoger si se desea integrar con algún otro *framework* o *librería* como Angular, Vue o incluso React; aunque puede utilizarse por separado sin ninguna de estas otras herramientas.

Se trata de un *framework* maduro, con una comunidad en continuo crecimiento y que ofrece constantemente soporte y nuevos plugins. A pesar de ser un software de código abierto, cuenta con un equipo completamente dedicado a su mantenimiento y desarrollo. Recientemente, se liberó la versión 4 la cual representa un antes y un después, cambiando la filosofía del *framework* en varios aspectos. Así pues, el equipo ha logrado trazar un camino claro de lo que se puede esperar en un futuro, dándole a las empresas y desarrolladores una mayor confianza para que puedan basar sus productos en esta herramienta.

Los objetivos principales del equipo de Ionic y su comunidad para el framework son claros⁸.

- Crear y desplegar aplicaciones que puedan trabajar sobre diferentes plataformas como IOS, Android, *Desktops*, y en la web como PWAs; todas bajo un mismo código base.
- “*Write once, run anywhere*”. - Ionic team.
- Desarrollar una herramienta basada en tecnologías web que estén estandarizadas y sean confiables (e.g. *Custom Elements* y *Shadow DOM* en el uso de HTML, CSS y JavaScript). Para así ser un *framework* completamente desligado de cualquier otro proveedor de alguna otra librería (e.g. Google y Angular).
- Brindar elementos listos para su uso *out-of-the-box*, que sean simples, funcionales y agradables a lo largo de las múltiples plataformas soportadas.
- Facilitar y mejorar la experiencia de desarrollo brindando una herramienta simple, que resulte agradable el desarrollo sobre esta y que pueda ser aprendida por prácticamente cualquiera que posea conocimientos de tecnología web.

5.2. Angular

Es un *framework* de código libre, igual que Ionic. Provee un conjunto de herramientas para desarrollar aplicaciones web tanto para móviles como para computadoras de escritorio, con la diferencia que Angular no provee la utilidad para generar aplicaciones móviles.

⁸ Ionic Team. *Ionic Framework*. <https://ionicframework.com/docs>.

Angular es uno de los *frameworks* más antiguos desarrollado por un equipo de Google en el 2010 y rescrito en el 2016. Actualmente, cuenta con casi 1 000 contribuidores, liderados siempre por un equipo de Google. Esto se refleja en el constante soporte que se le da a la herramienta. La última versión estable de Angular es la versión 8 recién liberada el 28 de mayo del presente año.

Tras la actualización del 2016, el equipo de Angular se ha enfocado en ganar la confianza de sus usuarios (i.e. desarrolladores individuales y compañías) estableciendo lineamientos claros en cuanto a la liberación de nuevas versiones y la duración del soporte para cada una. En general, se prevé la liberación de una versión principal (*major release*) cada 6 meses con un soporte de 18 meses. Esto refleja el compromiso y formalidad que el equipo tiene con sus usuarios y ayuda a las compañías y desarrolladores a planificar la revisión y actualización periódica de sus soluciones.

Angular al ser una solución completa provee un conjunto de herramientas para solucionar diferentes problemas. Angular ayuda en creación de componentes de UI, manejo de animaciones, aplicación de políticas de acceso, control de flujo de datos, navegación entre pantallas y más. Por eso, la curva de aprendizaje tiende a ser un poco más pronunciada pues para llegar a dominar la herramienta se debe entender varios conceptos afines.

En general, Angular provee los mecanismos para desarrollar una solución multiplataforma (usando librerías como Cordova o frameworks como ionic), de alto rendimiento; tratando de aumentar la productividad por medio de su CLI.

5.3. **Firestore**

Es un conjunto de servicios ofrecidos por Google para facilitar el desarrollo, despliegue y mantenimiento de aplicaciones. Entre los servicios que ofrece están *Cloud Firestore*, *Cloud Functions*, *Authentication*, *Hosting*, *Cloud Storage* y *Realtime Database*. Cada uno de ellos se proveen como *PaaS (Platform as a Service)*, por lo cual el desarrollador aún debe realizar configuraciones mínimas para su uso. Además, Firestore ofrece otra gama de herramientas que ayudan a mejorar la calidad de la aplicación por medio de monitoreos, reportes y pruebas.

Firestore ayuda a crear soluciones integrales con una infraestructura completamente *server-less*. Esto mejora la productividad ya que el desarrollo se centra en la solución y sus características y no en la configuración de servidores, redes, esquemas de seguridad. Además, ayuda a realizar una menor inversión inicial escalando los recursos requeridos según se requiera.

5.4. **Cloud Firestore**

Es una base de datos para aplicaciones móviles, web y para servidores. Es sucesora de *Realtime Database*, la cual ofrece mejoras en cuanto a su flexibilidad y escalabilidad entre otros. Igual que su predecesor, ofrece sincronización de datos en *tiempo real* a lo largo de los dispositivos suscritos y soporte para el manejo de datos fuera de línea tanto en móviles como en aplicaciones web. Además, se integra con servicios como *Cloud Functions* y *Firestore Authentication* para generar una solución integrada.

Esta base de datos es conveniente para las aplicaciones web desarrolladas con JavaScript pues su lenguaje se asemeja a la notación JSON. Por esto, su comunicación resulta mucho más natural, facilitando el desarrollo y

mantenimiento. Las unidades más granulares de datos son guardadas en documentos, estos a su vez se pueden agrupar en colecciones. Cada documento puede contener una serie de llaves que mapean a un valor e incluso a subcolecciones. Así se pueden generar estructuras de datos que tengan sentido para el sistema.

En general, es una base de datos NoSQL en la nube a la cual pueden acceder directamente las aplicaciones desde Android, IOS o la web sin arriesgar la seguridad ya que se puede integrar con *Firebase Authentication* y *Firestore Security Rules*, que se integra fácilmente a las aplicaciones web por las semejanzas que comparten en el modelo de datos que maneja y el lenguaje que utiliza para realizar las consultas.

5.5. Firebase authentication

Es un servicio que ofrece la oportunidad de autenticar a los usuarios que soliciten acceso por medio de diferentes mecanismos. Soporta la autenticación tradicional por medio de un nombre de usuario y contraseña, así como por números de teléfono y proveedores de identidad federada como Google, Facebook, Twitter y más. Además, posee librerías de IU ya elaboradas para autenticar al usuario según el método preferido dentro de nuestra aplicación.

Es un servicio que se puede integrar estrechamente con los demás servicios ofrecidos por Firebase, aplicando los estándares como OAuth 2.0 y OpenID Connect. Así pues, se puede configurar para que funcione con un proveedor de identidad personalizado.

Como se ha mencionado, existen diferentes maneras de implementar la autenticación dentro de una aplicación. A continuación, se describen los métodos disponibles para cada plataforma y los requerimientos de cada una.

Tabla III. **Métodos de autenticación Firestore**

Método	Descripción y requerimientos.
FirebaseUI Auth.	<p>Es una forma de autenticación directa. Requiere que el desarrollador realice configuraciones mínimas. Esta solución se recomienda porque contempla los casos de uso extremos como la recuperación de contraseña y vinculación de cuentas. Además, es de código abierto, así que, constantemente es revisada para que cumpla con los estándares de seguridad recomendados.</p> <p>Disponible para IOS, Android y Web.</p>
Autenticación basada en correo electrónico y contraseña.	<p>Este método utiliza el SDK para autenticar al usuario por medio del correo y la contraseña. También maneja el envío de correos para la recuperación de la contraseña.</p> <p>Disponible para IOS, Android, Web, C++ y Unity.</p>
Integración con proveedores de identidad federada.	<p>El SDK ofrece la oportunidad de integrarse con servicios de identidad externos como Google, Facebook, Twitter y Github.</p> <p>Disponibles para IOS, Android, Web, C++ y Unity.</p>
Autenticación por medio de un número telefónico.	<p>Envía mensajes de texto a un número telefónico para autenticarlos.</p> <p>Disponible para IOS, Android, Web, C++ y Unity.</p>

Continuación de la tabla III.

Integración con sistemas de autenticación personalizados.	El SDK también ofrece la flexibilidad para configurar un sistema de autenticación personalizado. Disponible para IOS, Android, Web, C++ y Unity.
Autenticación anónima.	Firebase Authentication contempla incluso la necesidad de crear cuentas anónimas para cuando se requiere brindar acceso temporal a un usuario sin requerir un registro formal. Una vez que el usuario decida registrarse, se puede convertir dicha cuenta en una cuenta normal de manera que pueda continuar con su uso ininterrumpidamente.

Fuente: elaboración propia.

Firebase cuenta con una gama de métodos para autenticar a un usuario. De esta manera, si se carece de un método establecido es posible agilizar el proceso de su elaboración. Si se cuenta con uno establecido, es posible seguir aprovechando con el beneficio adicional de interconectarse con los servicios de Firebase.

CONCLUSIONES

- El punto de venta desarrollado cumple con los requisitos necesarios para ayudar a las MIPYMEs en el proceso de captura, elaboración y cobro de sus platillos, brindándoles la oportunidad de ofrecer un servicio más eficiente y ordenado. Además, abre la puerta para un posterior análisis de ventas para elaborar decisiones más acertadas.
- Los *frameworks*, librerías y lenguajes que ofrecen un desarrollo multiplataforma, sea nativo o híbrido puede resultar en una buena alternativa al desarrollo nativo convencional, debido a la reducción de costos y tiempos.
- Ionic facilita el diseño de aplicaciones web y móviles, logra crear una interfaz elegante e intuitiva que ofrezca una experiencia similar a las aplicaciones nativas.
- Las infraestructuras serverless o en la nube resultan en una inversión inicial menor, más flexible y fácil de mantener. Esto permite que las empresas se centren en su giro de negocio dedicando un menor tiempo al acondicionamiento de un espacio dedicado, configuración y mantenimiento.
- Actualmente, existen varios certificadores para integrar el servicio de facturación electrónica en línea (FEL) en Guatemala que ofrecen una variedad de servicios fáciles de consumir como lo es el caso de INFILE SA.

RECOMENDACIONES

1. En la actualidad, los puntos de venta se enfocan en el negocio y no en el cliente. Desarrollan una solución donde involucrar al cliente en el proceso, puede significar una oportunidad para generar una mayor fidelidad por medio de ofertas personalizadas, mejores tiempos de servicio, entre otros.
2. Para el desarrollo de aplicaciones móviles se pide que los servicios con los que se comunicará sean servicios basados en estándares web por su estrecha relación, ya que el medio de comunicación será un canal web, esto puede generar un desarrollo más ágil y eficiente.
3. Al diseñar aplicaciones con un código base multiplataforma, se debe validar el diseño en cada una de las mismas y en los dispositivos que ofrecen ya que cada plataforma y cada versión puede presentar resultados inesperados por temas de compatibilidad en las tecnologías de fondo.
4. Al diseñar una arquitectura se debe considerar la inversión inicial, a mediano plazo y largo plazo. Además, se debe considerar la factibilidad y esfuerzo el momento de requerir escalarla. Para un desarrollo ágil para comenzar con una arquitectura *serverless* para obtener la mayor flexibilidad al menor costo posible.

BIBLIOGRAFÍA

1. Banco Mundial. *Desarrollo de PyMEs en Guatemala, facilitando el florecimiento de 10,000 empresas*, informe No. 54242-GT, 2010. 28 p. [en línea]. <<http://documents1.worldbank.org/curated/es/621881468251746633/pdf/542420v10ESW0S10Vol011Final1Espa1ol.pdf>> . [Consulta: 10 de agosto 2018].
2. Central Intelligence Agency. *The world factbook*. [en línea]. <<https://www.cia.gov/library/publications/the-world-factbook/rankorder/2001rank.html#gt>>. [Consulta: 10 de agosto 2018].
3. Instituto Nacional de Estadística INE. *Encuesta nacional de empleo e ingresos – ENEI, 2017*, 70 p.
4. Ionic Team. *Ionic framework*. [en línea] <<https://ionicframework.com>>. [Consulta: 15 de julio 2019].
5. Ministerio de Economía. *Sistema nacional de información MIPYME Guatemala, 2015*. [en línea]. <http://www.mineco.gob.gt/sites/default/files/Comunicacion%20Social/sistema_nacional_de_informacion_mipyme_guatemala_ano_base_2015.pdf>. [Consulta: 10 agosto 2018].
6. Naciones Unidas. *Country profiles*. [en línea]. <<https://esa.un.org/unpd/wup/DataQuery/>> . [Consulta: 10 de agosto 2019].

7. Oracle Hospitality. *Control de costes en el sector de la restauración*. [en línea]. <<https://www.oracle.com/assets/european-spanish-cost-ctrl-report-3206652.pdf>>. [Consulta: 5 de junio 2019].
8. SAT. *Nuevo régimen de factura electrónica en línea (acuerdo de directorio número SAT-13-2018)*. [en línea]. <<https://portal.sat.gob.gt/portal/descarga/1740/factura-electronica/25094/acuerdo-de-directorio-13-2018.pdf>>. [Consulta: 10 de agosto 2018].
9. STAFF, Verge. *IOS: A visual history. 2013*. [en línea]. <<http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>>. [Consulta: 23 de agosto 2018].
10. Statista. *Number of apps available in leading app stores*. [en línea]. <<https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>>. [Consulta: 23 de agosto 2018].
11. Statista. *Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018*. [en línea]. <<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>>. [Consulta: 23 de agosto 2018].