

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Presentado por:

JUAN CARLOS, PALOMINO MIRANDA

**APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA
GESTIÓN DE PEDIDOS DE COMIDA VÍA DELIVERY
EN LOS DISTRITOS DE ANDAHUAYLAS, TALAVERA
Y SAN JERÓNIMO**

Asesor:

Msc. JUAN JOSÉ ORÉ CERRÓN

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

ANDAHUAYLAS – APURIMAC – PERU

2019

APROBACIÓN DEL ASESOR



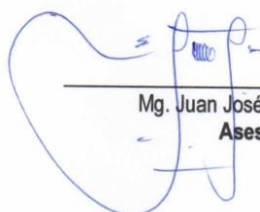
APROBACION DEL ASESOR

Quién suscribe:
Mg. Juan José Oré Cerrón por la presente:

CERTIFICA,

Que, el Bachiller en Ingeniería de Sistemas, JUAN CARLOS PALOMINO MIRANDA ha culminado satisfactoriamente el Proyecto de Tesis intitulado: "APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA GESTIÓN DE PEDIDOS DE COMIDA VÍA DELIVERY EN LOS DISTRITOS DE ANDAHUAYLAS TALAVERA Y SAN JERÓNIMO" para optar el Título Profesional de Ingeniero de Sistemas.

Andahuaylas, 14 de noviembre del 2019.



Mg. Juan José Oré Cerrón
Asesor



Br. Juan Carlos Palomino Miranda
Tesista

APROBACIÓN DEL JURADO DICTAMINADOR



APROBACIÓN DEL JURADO DICTAMINADOR

LA TESIS: "APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA GESTIÓN DE PEDIDOS DE COMIDA VÍA DELIVERY EN LOS DISTRITOS DE ANDAHUAYLAS TALAVERA Y SAN JERÓNIMO"; para optar el Título Profesional de Ingeniero de Sistemas, ha sido evaluada por el Jurado Dictaminador conformado por:

PRESIDENTA: DRA CECILIA EDITH GARCIA RIVAS PLATA

PRIMER MIEMBRO: DR. YALMAR TEMISTOCLES PONCE ATENCIO

SEGUNDO MIEMBRO: DR. JULIO CÉSAR HUANCA MARIN

Habiendo sido aprobado por **UNANIMIDAD**, en la ciudad de Andahuaylas el día 04 del mes de setiembre del 2019.

Andahuaylas, 14 de noviembre del 2019.

DRA. CECILIA EDITH GARCIA RIVAS PLATA
PRESIDENTE DEL JURADO DICTAMINADOR

DR. YALMAR TEMISTOCLES PONCE ATENCIO
PRIMER MIEMBRO DEL JURADO DICTAMINADOR

DR. JULIO CESAR HUANCA MARIN
SEGUNDO MIEMBRO DEL JURADO DICTAMINADOR

DECLARACIÓN JURADA DE AUTENTECIDAD

ANEXO 33



DECLARACIÓN JURADA DE AUTENTECIDAD

Yo, **Juan Carlos Palomino Miranda**, identificado con DNI N° 48111458 de la Escuela Profesional de Ingeniería de Sistemas.

Declaro bajo juramento que el Proyecto Titulado: **APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA GESTIÓN DE PEDIDOS DE COMIDA VÍA DELIVERY EN LOS DISTRITOS DE ANDAHUAYLAS, TALAVERA Y SAN JERÓNIMO.**

Es auténtico y no vulnera los derechos de autor. Además, su contenido es de entera responsabilidad del autor (es) del proyecto, quedando la UNAJMA exenta de toda responsabilidad en caso de atentar contra la Ley de propiedad intelectual y derechos de autor.

Andahuaylas, 14 de noviembre del 2019.

Firma

N° DNI:48111458

E-mail: admjc02@gmail.com

N° Celular: 928080743

ACTA DE SUSTENTACIÓN DE TESIS



Universidad Nacional José María Arguedas

Identidad y Excelencia para el Trabajo Productivo y el Desarrollo



FACULTAD DE INGENIERÍA


ACTA DE SUSTENTACIÓN DE TESIS


En la Av. José María Arguedas del Local Académico SL01 (Ccoyahuacho) en el auditorio de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas ubicado en el distrito de San Jerónimo de la Provincia de Andahuaylas, siendo las 09:00 horas del día 04 de setiembre del año 2019, se reunieron los docentes: Dra. Cecilia Edith García Rivas Plata, Dr. Yalmar Temistocles Ponce Atencio, Dr. Julio César Huanca Marín, en condición de integrantes del Jurado Evaluador del Informe Final de Tesis intitulado: "APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA GESTIÓN DE PEDIDOS DE COMIDA VÍA DELIVERY EN LOS DISTRITOS DE ANDAHUAYLAS, TALAVERA Y SAN JERÓNIMO", cuyo autor es el Bachiller en Ingeniería de Sistemas **JUAN CARLOS PALOMINO MIRANDA**, el asesor Mtr. Juan José Oré Cerrón, con el propósito de proceder a la sustentación y defensa de dicha tesis.

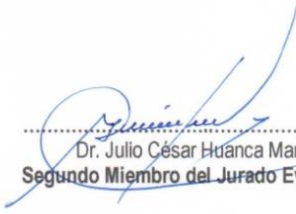
Luego de la sustentación y defensa de la tesis, el Jurado Evaluador **ACORDÓ: aprobar** por **unanimidad** al Bachiller en Ingeniería de Sistemas **JUAN CARLOS PALOMINO MIRANDA**, obteniendo la siguiente calificación y mención:

Nota escala vigesimal		Mención
Números	Letras	
17	diecisiete	muy bueno

En señal de conformidad, se procedió a la firma de la presente acta en 03 ejemplares.


Dra. Cecilia Edith García Rivas Plata
Presidente del Jurado Evaluador


Dr. Yalmar Temistocles Ponce Atencio
Primer Miembro del Jurado Evaluador


Dr. Julio César Huanca Marín
Segundo Miembro del Jurado Evaluador

DEDICATORIA

El presente trabajo de investigación lo dedico en primer lugar a Dios por estar siempre conmigo en cada paso que doy, por ser el inspirador y darme fuerzas para continuar; a mis padres por su amor incondicional, trabajo y sacrificio en todos estos años que gracias a ellos llegué hasta aquí; a mis hermanos por estar siempre presentes brindándome su apoyo moral en toda esta etapa.

AGRADECIMIENTO

En primer lugar, agradezco a mis docentes formadores de la UNIVERSIDAD NACIONAL JOSE MARÍA ARGUEDAS quienes fueron la guía y el apoyo para poder llegar al punto en el que me encuentro.

El desarrollo de esta tesis no lo puedo catalogar como algo fácil, pero lo que puedo afirmar, es que durante todo este tiempo pude disfrutar de cada momento, que cada investigación, proceso y proyectos que se realizaron dentro de esta, lo disfruté mucho, y no fue porque simplemente me dispuse a que así fuera, fue porque mis amigos siempre estuvieron ahí, fue porque la vida misma me demostró que de las cosas y actos que yo realicé, serán los mismos que harán conmigo.

ÍNDICE GENERAL

APROBACIÓN DEL ASESOR	II
APROBACIÓN DEL JURADO DICTAMINADOR	III
DECLARACIÓN JURADA DE AUTENTECIDAD	IV
ACTA DE SUSTENTACIÓN DE TESIS	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
1. PLANTEAMIENTO DEL PROBLEMA	20
1.1. Realidad problemática.....	20
1.2. Formulación del problema	21
1.3. Objetivos	22
1.3.1. Objetivo General.....	22
1.3.2. Objetivos Específicos	22
1.4. Justificación	22
1.5. Viabilidad de la investigación	22
1.5.1. Viabilidad Técnica.....	22
1.5.2. Viabilidad Económica	23
1.5.3. Viabilidad Social	24
1.5.4. Viabilidad Legal	24
1.6. Limitación del estudio.....	24
2. MARCO TEÓRICO	26
2.1. ANTECEDENTES INTERNACIONALES.....	26
2.2. ANTECEDENTES NACIONALES	26
2.3. BASES TEÓRICA	28
2.3.1. Aplicación Móvil.....	28
2.3.2. Aplicación Nativa	29
2.3.3. Aplicación Web.....	30
2.3.4. Aplicación Híbrida.....	30
2.3.5. Metodologías Ágiles para el Desarrollo de Software.....	31
2.3.6. Comparación Metodologías Ágiles y Tradicionales.....	38
2.3.7. Desarrollo de Aplicaciones para dispositivos Móviles	39
2.3.8. Sistemas Operativos para Dispositivos Móviles.....	40
2.3.9. Ionic.....	42
2.3.10. Cordova.....	42

2.3.11. Delivery	43
2.3.12. MySQL	43
2.3.13. API REST	44
3. PROPUESTA DE SOLUCIÓN	46
3.1. Ingeniería del Proyecto	46
3.2. Base de datos de la aplicación.....	46
3.3. Diseño Modular	47
3.4. Uso de metodologías de Programación Mobile-D	54
3.4.1. Fase de Exploración	54
3.4.2. Fase de Iniciación.....	57
3.4.3. Fase de Producción.....	65
3.4.4. Fase de Estabilización	79
3.4.5. Fase de Pruebas	101
3.5. Administración y Gestión del Proyecto Basado en PMBOOK.....	101
3.5.1. Plan de Gestión de la Integración	102
3.5.2. Plan de Gestión del Alcance.....	105
3.5.3. Plan de Gestión del Tiempo.....	106
3.5.4. Plan de Gestión de Costos	108
3.5.5. Plan de Gestión de Calidad	109
3.5.6. Plan de Gestión de Recursos Humanos	111
3.5.7. Plan de Gestión de Comunicaciones	112
3.5.8. Plan de Gestión de Riesgos	113
3.5.9. Plan de Gestión de Adquisiciones	114
3.5.10. Plan de Gestión de Los Interesados	116
4. CONCLUSIONES	118
5. RECOMENDACIONES.....	118
6. REFERENCIAS BIBLIOGRÁFICAS.....	119
7. ANEXOS.....	121

ÍNDICE DE TABLAS

Tabla N° 1:Diferencias entre metodologías ágiles y no ágiles	38
Tabla N° 2: Información del Proyecto.....	56
Tabla N° 3: Caso de Uso Iniciar Aplicación.....	58
Tabla N° 4: Caso de Uso Crear Nueva Cuenta.....	59
Tabla N° 5: Caso de Uso Recuperar Contraseña.....	59
Tabla N° 6: Caso de Uso Menú Principal	60
Tabla N° 7: Caso de Uso Seleccionar Establecimiento.....	60
Tabla N° 8: Caso de Uso Buscar Producto	61
Tabla N° 9: Caso de Uso Agregar Producto.....	61
Tabla N° 10: Caso de Uso Listar Pedido.....	62
Tabla N° 11: Caso de Uso Carrito de Compras	62
Tabla N° 12: Caso de Uso Cambiar Dirección	63
Tabla N° 13: Caso de Uso Generar Pedido	64
Tabla N° 14: Caso de Uso Comprar.....	64
Tabla N° 15: Estructura de la Aplicación Móvil.....	66
Tabla N° 16: Directorios del Proyecto PEDIDOS	70
Tabla N° 17: Tipos de archivos utilizados en el Proyecto.....	71
Tabla N° 18: Directorio app del Proyecto	72
Tabla N° 19: Directorio assets.....	73
Tabla N° 20: Directorio components.....	73
Tabla N° 21: Directorio core	74
Tabla N° 22: Directorio models.....	74
Tabla N° 23: Directorio pages	75
Tabla N° 24: Directorio services	78
Tabla N° 25: Plan de Gestión de la Integración	102
Tabla N° 26: Plan de Gestión del Alcance	105
Tabla N° 27: Plan de Gestión del Tiempo	106
Tabla N° 28: Plan de Gestión de Costos.....	108
Tabla N° 29: Plan de Gestión de Calidad.....	109
Tabla N° 30: Plan de Gestión de Recursos Humanos.....	111
Tabla N° 31: Plan de Gestión de Comunicaciones.....	112

Tabla N° 32: Plan de Gestión de Riesgos	113
Tabla N° 33: Plan de Gestión de Adquisiciones	114
Tabla N° 34: Plan de Gestión de los Interesados.....	116

ÍNDICE DE FIGURAS

Figura N° 1: Árbol de Problemas las posibilidades de un negocio de comidas se ven afectadas por la carencia de una herramienta tecnológica.....	21
Figura N° 2: Netflix tiene una aplicación híbrida que se ve prácticamente igual en iOS y en Android	31
Figura N° 3: Las prácticas se refuerzan entre si.....	34
Figura N° 4: Flujo de Proceso SCRUM	35
Figura N° 5: Ciclo de vida de la Metodología Mobile-D	38
Figura N° 6: Modelo Entidad Relación de la Aplicación	46
Figura N° 7: Módulo Inicio	47
Figura N° 8: Módulo Locales	48
Figura N° 9: Módulo Productos - Local.....	49
Figura N° 10: Módulo Login.....	50
Figura N° 11: Módulo Registro	51
Figura N° 12: Módulo Pedido	52
Figura N° 13: Módulo Comprar.....	53
Figura N° 14: Fases de la Metodología Mobile-D.....	54
Figura N° 15: Dependencias de la Aplicación	67
Figura N° 16: Plugins Necesarios para la Aplicación Móvil-Cordova	68
Figura N° 17: Configuración de app.module.ts.....	69
Figura N° 18:Estructura del Directorio PEDIDO	70
Figura N° 19: Directorio src	71
Figura N° 20: Directorio app	72
Figura N° 21: Directorio assets.....	73
Figura N° 22: Directorio components	73
Figura N° 23: Directorio core.....	74
Figura N° 24: Directorio models	74
Figura N° 25: Directorio pages	75
Figura N° 26: Directorio services.....	77
Figura N° 27: Página Raiz.....	80
Figura N° 28: login.html.....	81
Figura N° 29:login.ts.....	82

Figura N° 30: asignar-pedido.html.....	83
Figura N° 31: asignar-pedido.ts.....	85
Figura N° 32: cart.html	86
Figura N° 33: cart.ts	89
Figura N° 34: entrega.html	90
Figura N° 35: entrega.ts	91
Figura N° 36: home.html	92
Figura N° 37: home.ts	93
Figura N° 38: map.html	93
Figura N° 39: map.ts	94
Figura N° 40: producto.html	95
Figura N° 41: producto.ts	96
Figura N° 42: register.html	98
Figura N° 43: register.ts	99
Figura N° 44: search.html.....	99
Figura N° 45: search.ts.....	101
Figura N° 46: Acceso a la aplicación.....	121
Figura N° 47: Menú Principal de la aplicación.....	122
Figura N° 48: Categoría Pizzerías.....	122
Figura N° 49: Productos Establecimiento.....	123
Figura N° 50: Agregar Productos al Carrito de Compras	123
Figura N° 51: Carrito de Compras	124
Figura N° 52: Generar Pedido.....	125
Figura N° 53: Realizar pago para generar el pedido.	125
Figura N° 54: Web descarga Node.js	126
Figura N° 55: Pasos instalación Node.js	127
Figura N° 56: Instalación Node.js.....	127
Figura N° 57: Portal de descarga Visual Code Studio.....	128
Figura N° 58: Pasos Instalación Visual Code Studio.....	128
Figura N° 59: Acceso directo Visual Code Studio	129
Figura N° 60: Menú principal Visual Code Studio.....	129
Figura N° 61: Verificar Versión del Node.js	130

Figura N° 62: Instalación de Cordova.....	130
Figura N° 63: Instalación IONIC	130
Figura N° 64: Crear Proyecto Ionic.....	130

RESUMEN

El presente informe final del proyecto de tesis, tiene como objetivo principal el desarrollo de una aplicación móvil multiplataforma, para la gestión de pedidos de comida vía delivery en los distritos de Andahuaylas, Talavera y San Jerónimo, la cual llevará el nombre de "WASIMAN", para el desarrollo de esta aplicación móvil se utilizó diferentes herramientas de desarrollo tecnológico como IONIC, MySQL, NodeJS, Apache Cordova, estas herramientas proporcionan características únicas y especiales para que la aplicación móvil sea altamente funcional cumpliendo los requerimientos planteados al inicio del proyecto.

El framework que se utilizó fue IONIC, que está basado en JavaScript, HTML, y CSS, esto permite que las aplicaciones desarrolladas en este framework puedan ser compiladas para diferentes plataformas móviles.

En el desarrollo de esta aplicación se hizo uso de una API REST, un conjunto de restricciones con las que podemos crear un estilo de arquitectura software, la cual podremos usar para crear aplicaciones web respetando HTTP. Esto permitió que se pueda crear servicios web las cuales tienen ciertas restricciones.

La metodología utilizada para el desarrollo de este proyecto fue la Metodología Mobile-D, que es una metodología que está basada en diversas tecnologías como Rational Unified Process, Extreme Programming y Crystal Methodologies, y la finalidad de esta metodología es obtener pequeños ciclos de desarrollo de forma rápida.

El costo del proyecto en la fase de producción de la aplicación móvil no tiene costo ya que aún no es necesario contratar un hosting.

Palabras Clave: IONIC, API REST, Mobile-D.

ABSTRACT

This final report of the thesis project, has as main objective the development of a multiplatform mobile application, for the management of food orders via delivery in the districts of Andahuaylas, Talavera and San Jerónimo, which will be called “WASIMAN” , for the development of this mobile application, different technological development tools were used such as IONIC, MySQL, NodeJS, Apache Cordova, these tools provide unique and special features so that the mobile application is highly functional fulfilling the requirements set at the beginning of the project.

The framework that was used was IONIC, which is based on JavaScript, HTML, and CSS, this allows the applications developed in this framework to be compiled for different mobile platforms.

In the development of this application, a REST API was used, a set of restrictions with which we can create a software architecture style, which we can use to create web applications respecting HTTP. This allowed web services to be created which have certain restrictions.

The methodology used for the development of this project was the Mobile-D Methodology, which is a methodology that is based on various technologies such as Rational Unified Process, Extreme Programming and Crystal Methodologies, and the purpose of this methodology is to obtain small development cycles of quick way.

The cost of the project in the production phase of the mobile application has no cost since it is not yet necessary to hire a hosting.

Keywords: IONIC, REST API, Mobile-D.

CHUMASQA

Kay proyecto de tesis rurasqamanta willakuy tukuy, kanmi chiqap puririyinín móvil multiplataforma rurayninkunapi qispichinapaq, llapa mikuykuna delivery chaynintakama Andahuaylas, Talavera y San Jerónimo mañakunanpaq, chaypa sutinmi “WASIMAN” kanqa, Kay ruray aplicación móvil qispichinapaq tukuy niraq llamkanakunawan IONIC, MySQL, NodeJS, Apache Cordova nisqankunawan hapipakuspaku yanapachikusqa, kay llamkanakunaqa huk llasaq riqsichikuyllatam qukun chay aplicación movil nisqan aswan allin puririnanpaq chay proyectopa qallariyninpi tukuy kamachikuynin allin qispinanpaq.

Framework nisqanta IONIC paq churarqa, chaymi JavaScript, HTML, y CSS tupachisqa kaykuna qatinqa, framework niqanpas rurayninpi yanapanqa kaykunaqa akllasqa kasqa tukuy ima platafoma móviles kananpaq.

Rurayninpi kay API REST nisqanwan yanapachikurqa, kaywanqa huk huñusqa arquitectura software nisqantapas qispirachichuwanmi, chaynallataq huk ruraykunata HTTP nisqanta yupaychaspalla musuq aplicaciones WEB nisqankunata qispichichuwan.

Kay ruraypa llamkay purichiqninqa Mobile-D nisqanmi, kayqa tarikun tukuy ima kunan pacha llapa tecnologías nisqan ukupi kayhinakuna Rational Unified Process, Extreme Programing y Crystal Methodologies, kay ruraypa puririchiyinnqa chiqap yachachikuyninmi utqayman uchuy llamkaykunata tarinapaq purichinapaq ima.

Kay proyectopaqa ña llamkayninpi rantipiña kaspapas manam chaninqa kanchu, chaymi manaraq hosting nisqantaqa maskanachu.

Chiqap rimay: IONIC, API REST, Mobile-D.

INTRODUCCIÓN

Hoy en día el uso de Smartphones (teléfonos inteligentes), ha ido creciendo de manera rápida y de la misma manera, el uso de aplicaciones móviles se ha convertido en herramientas de gran valor que permite a los usuarios realizar muchas actividades, con ello las tecnologías que permiten su desarrollo también han evolucionado, mejorando e innovando día a día, esto obliga a los desarrolladores de aplicaciones móviles a estar actualizados en el ámbito de las tecnologías que permiten su desarrollo.

En esta tesis se detalla el uso del framework IONIC que como resultado genera una aplicación móvil multiplataforma sin costo en la implementación en la fase de desarrollo ni en la fase de producción, todo esto con la ventaja de ser escalable.

El resto del documento está organizado en cuatro capítulos:

- **Planteamiento del problema:** En este capítulo se menciona la realidad problemática, esto nos lleva al desarrollo y solución ante el problema identificado, permitiendo plantear nuestros objetivos general y específico, justificación y viabilidad para la ejecución.
- **Marco teórico:** En este capítulo haremos mención de todos los conceptos relacionados con el Análisis Diseño e Implementación de una aplicación móvil multiplataforma para la gestión de pedidos de comida vía delivery en los distritos de Andahuaylas, Talavera y San Jerónimo, de la misma manera encontraremos los antecedentes que se tomaron como fuente de iniciación para la investigación.
- **Materiales y Métodos:** En este capítulo se detallan de manera concisa el análisis, diseño e implementación de una aplicación móvil multiplataforma para la gestión de pedidos de comida vía delivery en los distritos de Andahuaylas, Talavera y San Jerónimo, utilizando la metodología Mobile-D la cual se adapta de mejor manera para el desarrollo del proyecto.
- **Resultados y Discusión:** En este capítulo se analiza el correcto funcionamiento de la aplicación móvil multiplataforma para la gestión de

pedidos de comida vía delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Realidad problemática

Andahuaylas es una de las 7 provincias de la región Apurímac y cuenta con una población estimada de 50,000 habitantes al 2018 según información del Instituto Nacional de Estadística e Informática (INEI).

En la población Andahuaylina, se puede observar que la gran mayoría tiene acceso a un celular inteligente (Smart Phone) el cual permite acceder a ciertas herramientas y aplicaciones que sean de su interés. En esta ciudad una gran cantidad de establecimientos se dedican a brindar el servicio de comida de las cuales son muy pocas las que ofrecen el servicio de delivery de los productos que ofrecen a sus clientes.

El crecimiento económico de la provincia de Andahuaylas, significo la apertura de restaurantes, pollerías, cevicherías, pizzerías y recreos turísticos, estos centros gastronómicos en su mayoría atienden utilizando toda su capacidad de aforo en horas punta, se ha podido notar en fechas especiales que las capacidades de atención han sido rebasadas por los clientes debido a que los locales tienen capacidades limitadas de atención por la infraestructura que poseen, existiendo demanda insatisfecha. Pero además muchos clientes potenciales desean disfrutar de una comida preparada en estos centros gastronómicos en la comodidad de su hogar. Pero de quince establecimientos de expendio de comidas solo dos establecimientos tienen el servicio de Delivery, este factor nos da una interesante oportunidad de negocio pero que debe ser complementado con un desarrollo tecnológico, que responde al siguiente árbol de problemas.

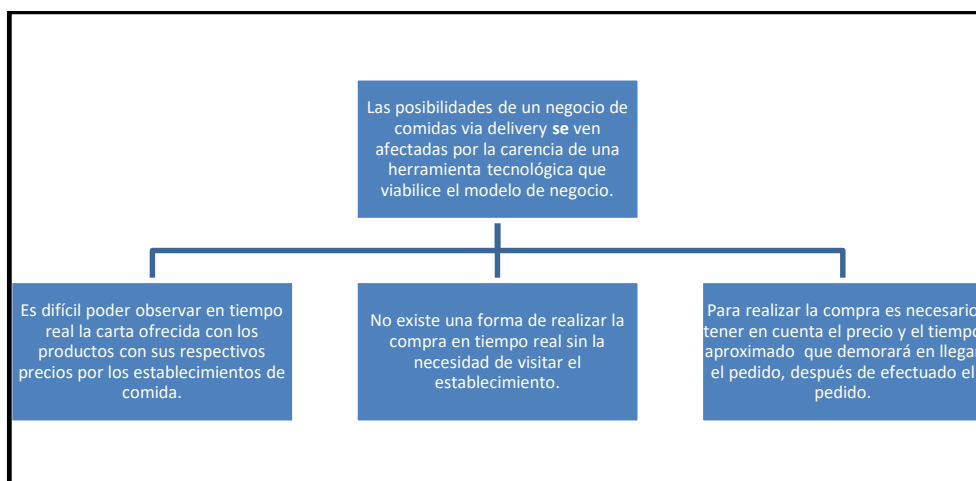


Figura N° 1: Árbol de Problemas las posibilidades de un negocio de comidas se ven afectadas por la carencia de una herramienta tecnológica.

Fuente Elaboración Propia.

(Hott Vidal & Toro Oyorzún, 2011) Menciona que; Hoy en día, las telecomunicaciones han logrado penetrar absolutamente en todas las áreas de la sociedad, tanto en el diario vivir de una familia como en la manera de realizar negocios de una empresa. Desde hace algunas décadas, las telecomunicaciones han sido utilizadas para apoyar y colaborar en la gestión de los negocios; el fax y el teléfono fijo fueron y son, herramientas que ayudaron a disminuir increíblemente los tiempos de todos los procesos relacionados con la comunicación entre diferentes entidades comerciales. Luego con la aparición del teléfono celular, muchas empresas se vieron inmensamente beneficiadas al poder contactarse con sus trabajadores que estaban fuera de las oficinas y así poder entregarles las directrices necesarias para sus labores.

1.2. Formulación del problema

Las posibilidades de un negocio de comidas vía Delivery, en los distritos de Andahuaylas, Talavera de la reina y San Jerónimo, se ven afectadas

por la carencia de una herramienta tecnológica que viabilice el modelo de negocio.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una aplicación móvil multiplataforma para la gestión de pedidos de comida en los distritos de Andahuaylas, Talavera de la reina y San Jerónimo.

1.3.2. Objetivos Específicos

- Desarrollar un módulo que permita mostrar la carta ofrecida por cada establecimiento.
- Desarrollar una aplicación móvil multiplataforma que permita gestionar comidas vía delivery utilizando un teléfono inteligente.
- Se desarrollará una función agregar producto para posteriormente el usuario pueda listar pedidos donde se de la información del precio total de los productos seleccionados.

1.4. Justificación

En la actualidad en los distritos de Andahuaylas, Talavera, San Jerónimo hay una preocupación por parte de los consumidores potenciales que desean degustar de los productos de los establecimientos de comida en la comodidad de su hogar, ya que no se cuenta con un sistema de gestión de pedidos de comida vía delivery, se desea que con el desarrollo de la aplicación móvil se pondrá fin a este problema ya que por medio de la aplicación los clientes tendrán acceso a la lista completa de productos ofrecidos por los establecimientos de comida, para luego ellos poder realizar el pedido correspondiente.

1.5. Viabilidad de la investigación

1.5.1. Viabilidad Técnica

1.5.1.1. Herramientas

Las herramientas que se mencionan a continuación son las más aptas para el desarrollo del proyecto.

- Node.Js v10.15.3
- Cordova
- Ionic
- MySQL
- Visual Studio Code
- Conexión a Internet
- Navegadores google Chrome y Mozilla Firefox

1.5.1.2. Equipos

- Laptop Toshiba Satélite Intel Core i7
- Dispositivo Móvil Samsung J5 Prime
- Dispositivo Móvil Iphone 8

1.5.1.3. Metodología

Mobile-D, es una metodología de desarrollo de software ágil, tiene como objetivo el desarrollo y gestión de proyectos con mucha rapidez, eficacia, flexibilidad y control.

1.5.2. Viabilidad Económica

Para la realización de este proyecto, se usarán herramientas libres, como Ionic que es un framework que permite el desarrollo de aplicaciones móviles híbridas de manera gratuita, Node.JS que es un entorno de ejecución de JavaScript orientado a eventos asíncronos, este está diseñado para lograr desarrollar aplicaciones en red que sean escalables, MySQL que es un sistema de gestión de base de datos relacional de código abierto, Cordova que es un marco de desarrollo móvil de código abierto que permite utilizar estándares web como HTML5 y CSS3 y JavaScript para desarrollo

multiplataforma, de igual manera se utilizarán navegadores como google Chrome y Firefox que son libres y gratuitos, Visual Studio Code que es un editor de texto plano gratuito.

Los dispositivos Hardware (laptop, Smartphone) son propios del investigador.

1.5.3. Viabilidad Social

El proyecto tiene como finalidad brindar un servicio para la gestión de pedidos de comida vía delivery al cliente mediante un aplicativo móvil, este aplicativo móvil mostrará los diferentes productos que ofrecen los establecimientos de comida en los distritos de Andahuaylas, Talavera y San Jerónimo.

El aplicativo móvil podrá ser utilizado por los pobladores que deseen realizar un pedido del establecimiento de comida que le sea conveniente dentro de los distritos de Andahuaylas, Talavera, San Jerónimo.

1.5.4. Viabilidad Legal

En este proyecto las herramientas que serán utilizadas son gratuitas y libres, por lo tanto, el proyecto es viable en todos los aspectos.

1.6. Limitación del estudio

- La aplicación móvil está orientado a personas que dispongan de un Smart Phone (Celular Inteligente).
- En el caso de los usuarios del sistema operativo Android, la versión mínima requerida será el 4.0.
- Es necesario que el usuario tenga conexión a internet en su Smart Phone ya que los datos que requiere la aplicación estarán alojados en la nube.

- Los usuarios deben brindar información real al momento de registrarse en la aplicación, de lo contrario esto generará inconvenientes.

2. MARCO TEÓRICO

2.1. ANTECEDENTES INTERNACIONALES

(Cajilima Alvarado, 2015) en su trabajo de investigación “DESARROLLO DE UNA APLICACIÓN, PARA DISPOSITIVOS MÓVILES QUE PERMITA ADMINISTRAR Y CONTROLAR RUTAS DE LOS VENDEDORES, APLICADA A LA EMPRESA "ALMACENES JUAN ELJURI CÍA.LTDA." División Perfumería”, menciona que, hoy en día un gran número de compañías, sobre todo las del sector de ventas se ven impulsadas a tener o desarrollar su propia aplicación móvil para sacar ventaja ante la competencia y así llegar a muchos otros mercados y agilizar sus procesos. Los usuarios necesitan obtener información de su empresa en tiempo real y en cualquier parte ya sea para consultas de stock y precios de productos, leer y responder un correo electrónico, o conversar directamente con algún cliente o proveedor y no depender de un computador con acceso a internet para obtener esta información.

2.2. ANTECEDENTES NACIONALES

(Rojas Alegre, 2018) en su tesis “APLICACIÓN MÓVIL PARA OPTIMIZAR LA BÚSQUEDA DE PRODUCTOS Y SERVICIOS UTILIZANDO GEOLOCALIZACIÓN EN LA CIUDAD DE HUARAZ EN EL 2018” concluyó que con el desarrollo de la aplicación se optimizó la búsqueda de productos y servicios utilizando geolocalización, los resultados obtenidos fueron muy positivos indicando ello. La aplicación móvil permitió difundir la ubicación de lugares donde adquirir los productos y servicios que el usuario requiere.

(Huamantingo Navarro, 2018) en su trabajo de investigación “APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA PUBLICACIÓN DE ANUNCIOS DE OFERTA Y DEMANDA LABORAL EN TIEMPO REAL” menciona lo siguiente: que hoy en día el uso del internet es indispensable, las tecnologías que se usan ahora son robustas, gracias

a las investigaciones y aportes de grandes desarrolladores a nivel mundial y ahora está a nuestro alcance.

(Daneri Beltrán, 2017) en su tesis que lleva como título “UN FRAMEWORK ÁGIL PARA REALIZAR PRUEBAS FUNCIONALES DE SOFTWARE ORIENTADO A APLICACIONES MÓVILES, CASO: APLICACIÓN MÓVIL PARA TRANSFERENCIAS BANCARIAS DESARROLLADA BAJO METODOLOGÍA SCRUM” Menciona que las realizaciones de pruebas sobre aplicaciones móviles son mucho más complejas que las pruebas realizadas para aplicaciones web o de escritorio, ya que existen diferentes factores que amplían drásticamente el espectro de pruebas.

Al seguir metodologías ágiles se deben realizar actividades de control de calidad en cada funcionalidad construida. Idealmente cada funcionalidad debería ser completada en una o dos semanas, y ello evidencia que se tienen tiempos muy cortos para garantizar su calidad.

2.3. BASES TEÓRICA

2.3.1. Aplicación Móvil

2.3.1.1. Definición

Una aplicación móvil, más comúnmente conocida como una aplicación, es un tipo de software de aplicación diseñado para ejecutarse en un dispositivo móvil, como un teléfono inteligente o una tableta. Las aplicaciones móviles con frecuencia sirven para proporcionar a los usuarios servicios similares a los que se accede en las PC. Las aplicaciones son generalmente unidades de software pequeñas e individuales con función limitada. Este uso del software de aplicación fue popularizado originalmente por Apple Inc. y su App Store, que ofrece miles de aplicaciones para iPhone, iPad y iPod Touch. (Techopedia, 2017)

2.3.1.2. Características

Según (Fernández Luna , Decsai, 2006), las características de los dispositivos móviles son:

- Son aparatos pequeños, con algunas capacidades de procesamiento.
- Con conexión permanente o intermitente a una red.
- Con memoria limitada.
- Diseñados específicamente para una función, pero que pueden llevar a cabo otras más generales.
- Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
- La mayoría de estos aparatos pueden ser transportados en el bolsillo del propietario y otros están integrados dentro de

otros mayores, controlando su funcionalidad (como puede ser el ordenador integrado en una lavadora).

2.3.1.3. Tipos de Dispositivos móviles

En la actualidad podemos los tipos de dispositivos móviles son:

Teléfono móvil. Los teléfonos móviles son de los aparatos sofisticados que encontramos en nuestro cotidiano quehacer. Para comprimir y descomprimir señales digitales codificadas, tienen que procesar millones de cálculos por segundo. (Fernández Luna , Decsai , 2004)

PDA's. Un Asistente Digital Personal ofrece al usuario calendarios, agendas, block de notas son sus características principales que en un futuro estos reemplazarán a las agendas clásicas que existen, estos también permiten descargar archivos, correo electrónico y materiales siempre y cuando estén conectado a internet. (Fernández Luna , Decsai , 2004)

Consolas. Estos dispositivos son denominados como “dispositivos orientados para juegos” ya que estos dispositivos son caracterizados por tener una gran capacidad de procesamiento de gráficos, de igual manera que los PDA's son capaces de reproducir archivos multimedia, cuentan con un navegador de internet y otros.

2.3.2. Aplicación Nativa

(Cuello & Vittone, 2013) Las aplicaciones nativas son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamados genéricamente Software Development Kit o SDK. Así, Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK.

Por una de las ventajas al desarrollar aplicaciones nativas, es que se puede aprovechar el uso del sistema de notificaciones del mismo sistema operativo para poder mostrar notificaciones al usuario, aunque no se esté utilizando la aplicación.

2.3.3. Aplicación Web

(Santiago Campión, Tralbaldo, Kamijo, & Fernández, 2015) Las aplicaciones basadas en la web son aquellos recursos y herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En ese sentido, las web apps no son más que una versión de la página web optimizada para su correcta visualización en dispositivos móviles. Al ser documentos web se basan en los lenguajes HTML5 y CSS3, y son accesibles desde el navegador de cualquier dispositivo, independientemente del sistema operativo que utilice.

Una de las grandes desventajas de las web apps, es que siempre se necesitará disponer de una conexión a internet para acceder a los contenidos.

2.3.4. Aplicación Híbrida

(Cuello & Vittone, 2013) Este tipo de aplicaciones es una especie de combinación entre las dos anteriores. La forma de desarrollarlas es parecida a la de una aplicación web – usando HTML, CSS y Javascript-, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es como si tratara de una aplicación nativa.

Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuirlas en cada una de sus tiendas.

A diferencia de las aplicaciones web, éstas permiten acceder, usando librerías, a las capacidades del teléfono, tal como lo haría una app nativa.



Figura N° 2: Netflix tiene una aplicación híbrida que se ve prácticamente igual en iOS y en Android

Fuente (Cuello & Vittone, 2013)

2.3.5. Metodologías Ágiles para el Desarrollo de Software

Las metodologías ágiles en general se catalogan según su enfoque y características esenciales, surgen como una alternativa a las metodologías tradicionales, y están basados en un modelo de desarrollo iterativo cuyo objetivo es capturar mejor los requisitos cambiantes y la gestión de riesgos, de esta manera las metodologías ágiles de desarrollo rompen el proyecto en iteraciones de diferentes longitudes, cada una de ellas generando un producto completo y entregable; e incremental donde el

producto final se va construyendo de bloque en bloque durante todo el ciclo de vida de desarrollo del producto. (Amaya Balaguera, 2013)

2.3.5.1. Metodología Extreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Letelier & Penadés, 2005)

2.3.5.1.1. Prácticas XP

Esta metodología pone a disposición las siguientes prácticas para que el equipo de desarrollo los ejecute.

- **El juego de la planificación:** Es un espacio habitual de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas:** La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema, pero sí que constituyan un resultado de valor para el negocio.

- **Metáfora:** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.
- **Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.
- **Pruebas:** La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización:** La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.
- **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores.
- **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

- **Integración continua:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente.
- **40 horas por semana:** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas.
- **Cliente in-situ:** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio.
- **Estándares de programación:** Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

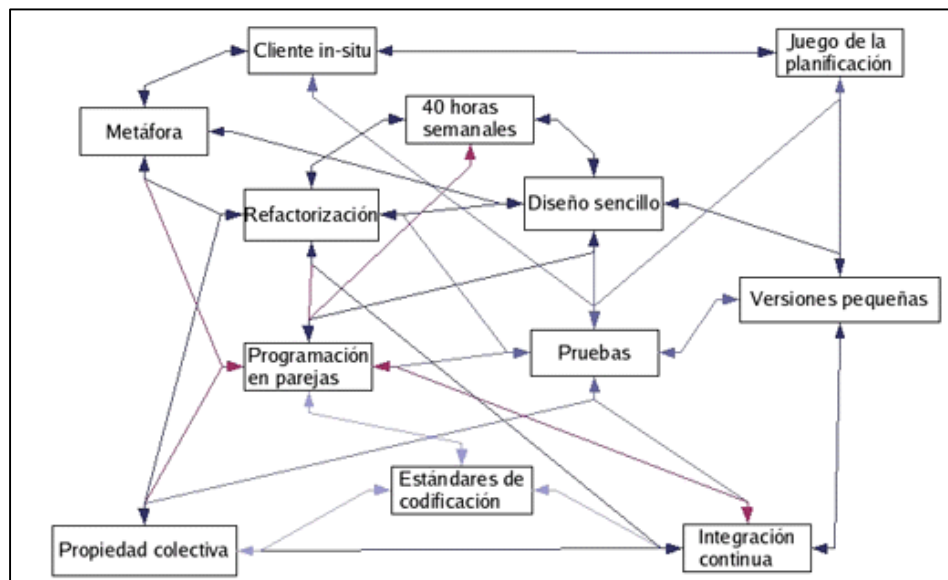


Figura N° 3: Las prácticas se refuerzan entre si

Fuente: (Letelier & Penadés, 2005)

2.3.5.2. SCRUM

Scrum se centra la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la

planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. (Amaya Balaguera, 2013)

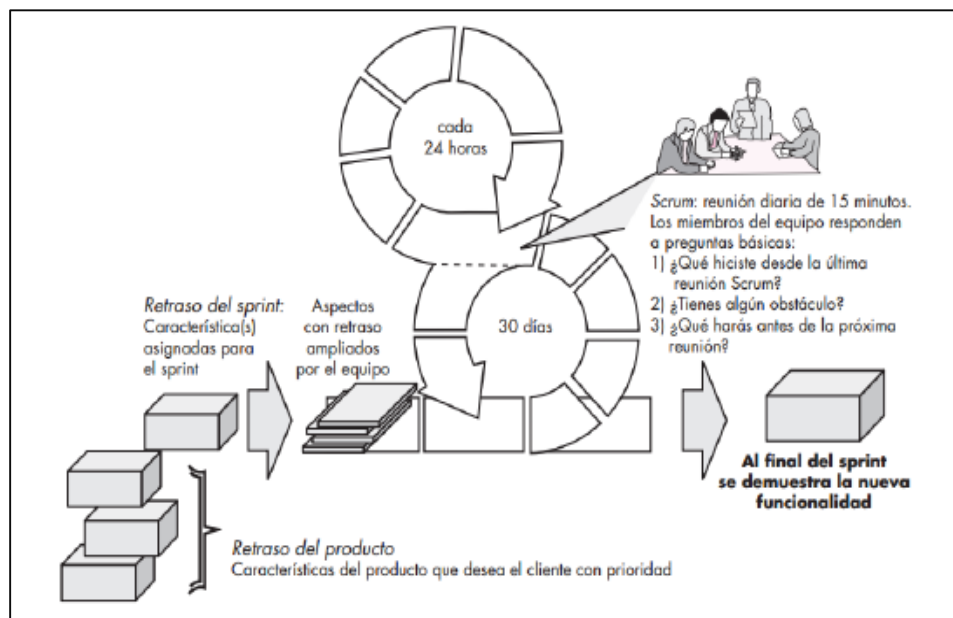


Figura N° 4: FLUJO de Proceso SCRUM

Fuente: (Atienza Pogo, 2017)

2.3.5.2.1. Roles SCRUM

La metodología SCRUM define roles y responsabilidades asignados a los procesos de desarrollo:

- **Dueño del Proyecto:** Se asegura de que el proyecto se esté desarrollando acorde con la estrategia del negocio. Escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

- **Master Scrum o Facilitador:** Elimina los obstáculos que impiden que el equipo cumpla con su objetivo.
- **Equipo de Desarrollo:** Los encargados de crear el producto para que pueda estar listo con los requerimientos necesarios.

2.3.5.2.2. Beneficios de la Metodología SCRUM

- Cumplimiento de expectativas.
- Flexibilidad a cambios.
- Reducción del Time to Market.
- Mayor Calidad de Software.
- Mayor Productividad.
- Maximiza retorno de la Inversión.
- Predicciones de Tiempos.
- Reducción de Riesgos.

2.3.5.3. Metodología Mobile-D

Es una metodología ágil la cual permite el desarrollo de aplicaciones móviles en tiempo record, es una de las pocas metodologías que cuenta con certificación CMMI (Capability Maturity Model Integration) que están orientadas al desarrollo de SDM (Marco de desarrollo de Software).

Se puede decir que Mobile-D es una creación un tanto antigua, ya que se desarrolló como parte del proyecto finlandés, ICAROS en el año 2004, fue creado mediante un proyecto de cooperación muy estrecha con la industria. El grueso del trabajo fue realizado por los investigadores del VTT. Aun así, la metodología de diseño se elaboró con una participación importante de las empresas de TI finlandesas.

Esta metodología es una mezcla de muchas técnicas, pero que al mismo tiempo lograron crear una contribución original para el desarrollo de nuevas aplicaciones para dispositivos móviles. (Blanco, Camarero, Fumero, Warterski, & Rodríguez, 2009)

Esta metodología tiene 5 fases:

- **EXPLORACIÓN:** En esta fase se lleva a cabo la planificación y planteamiento de requisitos principales del proyecto, también se definirá el objetivo general, alcance del proyecto y definición de los actores.
- **INICIACIÓN:** En esta fase los desarrolladores preparan y definen los recursos y herramientas necesarios que serán utilizados durante el desarrollo de la aplicación. De la misma manera se debe preparar el proyecto para evitar problemas que puedan surgir durante el desarrollo de la aplicación.
- **PRODUCCIÓN:** En esta fase se lleva a cabo toda la implementación de la aplicación móvil utilizando un ciclo de desarrollo iterativo e incremental, en el cual se usa el desarrollo dirigido por pruebas (TDD, Test-Driven-Development).
- **ESTABILIZACIÓN:** Se llevan a cabo las últimas acciones de integración para que de esta manera asegurar que el sistema funcione de manera correcta.
- **PRUEBA DEL SISTEMA:** Cuando el proyecto ya está concluido en su totalidad se procede a la fase de pruebas, en esta fase se deberá hacer pruebas de manera iterada hasta llegar a una versión estable de acuerdo a los requerimientos establecidos.

2.3.5.3.1. Ciclo de vida Metodología Mobile-D

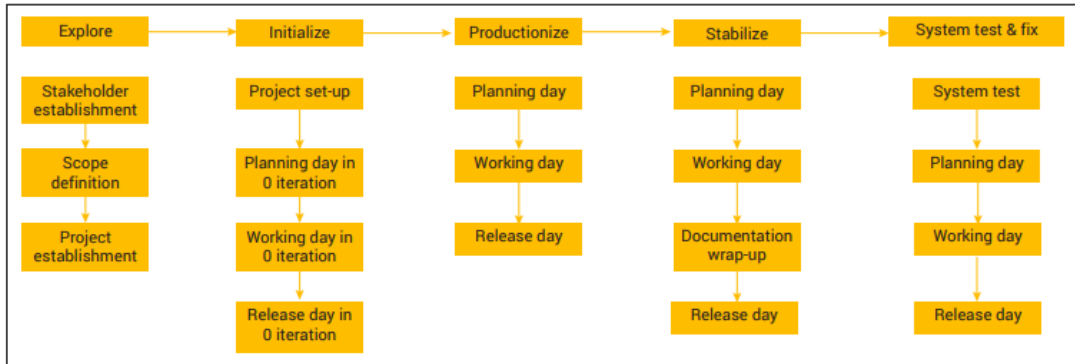


Figura N° 5: Ciclo de vida de la Metodología Mobile-D

Fuente: (Amaya Balaguera, 2013)

2.3.6. Comparación Metodologías Ágiles y Tradicionales

En la siguiente tabla se muestra las diferentes diferencias entre metodologías ágiles y tradicionales.

Tabla N° 1: Diferencias entre metodologías ágiles y no ágiles

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Pocos Artefactos. El modelado es prescindible, modelos desechables.
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o	Aplicables a proyectos de cualquier tamaño, pero suelen

entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Fuente: (Letelier & Penadés, 2005)

2.3.7. Desarrollo de Aplicaciones para dispositivos Móviles

El mercado de las aplicaciones móviles ha experimentado una expansión a pasos agigantados en los últimos 10 años, las plataformas móviles siguen mejorando su desempeño, y junto a ello la necesidad de los usuarios de que exista una gran variedad de aplicaciones.

El desarrollo de software para las plataformas móviles viene con características únicas y restricciones que se aplica a la mayoría de las etapas del ciclo de vida.

2.3.8. Sistemas Operativos para Dispositivos Móviles.

Los sistemas operativos con mayor influencia en el mercado de los dispositivos móviles son:

- Android
- iOS Apple
- Windows Phone
- BlackBerry

2.3.8.1. Android de Google

El sistema operativo Android es sin duda el líder del mercado móvil en sistemas operativos, está basado en Linux, diseñado originalmente para cámaras fotográficas profesionales, luego fue vendido a Google y modificado para ser utilizado en dispositivos móviles como los teléfonos inteligentes y luego en tablets como es el caso del Galaxy Tab de Samsung , actualmente se encuentra en desarrollo para usarse en netbooks y PC, el desarrollador de este SO es Google, fue anunciado en el 2007 y liberado en el 2008; además de la creación de la Open Handset Alliance, compuesto por 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para celulares, esto le ha ayudado mucho a Google a masificar el S.O, hasta el punto de ser usado por empresas como HTC, LG, Samsung, Motorola entre otros.

2.3.8.2. IOS de Apple

iOS es el sistema operativo que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV. Su simplicidad y optimización son sus pilares para que millones de usuarios se decanten por iOS en lugar de escoger otras plataformas que necesitan un hardware más potente para mover con fluidez el sistema operativo. Cada año, Apple lanza una gran

actualización de iOS que suele traer características exclusivas para los dispositivos más punteros que estén a la venta en ese momento.

2.3.8.3. Windows Phone de Microsoft

Windows Phone (abreviado WP) es un sistema operativo móvil desarrollado por Microsoft, como sucesor de Windows Mobile. Con Windows Phone; Microsoft ofrece una nueva interfaz de usuario que integra varios de sus servicios propios como OneDrive, Skype y Xbox Live en el sistema operativo. Compete directamente contra Android de Google e iOS de Apple. Debido a la evidente fragmentación de sus sistemas operativos, Microsoft anunció en enero de 2015 que dará de baja a Windows Phone, para enfocarse en un único sistema más versátil denominado Windows 10 Mobile, disponible para todo tipo de plataformas (teléfonos inteligentes, tabletas y computadoras). Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente y existe una gran oferta de software de terceros disponible para Windows Phone, la cual se puede adquirir a través de la tienda en línea de Windows Store para móviles.

2.3.8.4. BlackBerry OS de BlackBerry

BlackBerry es un sistema desarrollado por Research In Motion el cual fue presentado en el WES 2010 junto con un video promocional donde se muestra algunas novedades. RIM apuesta que su BlackBerry 6 estará enfocado en el mercado corporativo y no-corporativo. La mejor experiencia de este sistema se encontrará en los equipos touchscreen (Pantalla Táctil).

2.3.9. Ionic

Ionic es un framework gratuito y open source para desarrollar aplicaciones híbridas multiplataforma que utiliza HTML5, JavaScript, CSS (generado por SASS) y Cordova como base. Es uno de los framework del momento por utilizar AngularJS y TypeScript para gestionar las aplicaciones, lo que asegura aplicaciones rápidas y escalables. Es la plataforma de tecnología de desarrollo móvil multiplataforma más popular del mundo, impulsando el rápido crecimiento de las startups de algunas de las mayores compañías del mundo. Fue fundada en el 2012, cuando el uso de tecnologías web para construir aplicaciones nativas recién empezaba a surgir. (Ionic Framework, 2017)

2.3.9.1. Características

Ionic Presenta las siguientes características:

- **Alto rendimiento:** La velocidad es muy importante en las aplicaciones, por eso Ionic está desarrollado para que sea rápido gracias al Shadow DOM que es una API de aislamiento y ubicación.
- **Diseño Elegante:** Ionic fue diseñado para que pueda trabajar con todos los dispositivos móviles que existen en la actualidad.
- **Licencia de Código Abierto:** Ionic es de código abierto bajo licencia de MIT, esto significa que puede ser utilizado en proyectos personales o comerciales de manera gratuita.

2.3.10. Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como

HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo en cada plataforma móvil. (Apache Cordova, 2018)

2.3.11. Delivery

Se llama delivery al servicio de reparto que ofrece un comercio para entregar sus productos en el domicilio del comprador.

Actualmente el delivery también puede solicitarse a través de Internet. Desde un formulario en un sitio web hasta una aplicación en un dispositivo móvil, existen distintas herramientas digitales que posibilitan la solicitud de delivery.

Supongamos que un joven que vive solo no tiene ganas de cocinar y toma la decisión de pedir un delivery de pizza. El muchacho ingresa al sitio web de su pizzería favorita y, a través de un software, selecciona su menú: una pizza grande con jamón y tomate. El sistema le informa que, en media hora, recibirá su pedido. Cumplido ese plazo de tiempo, un empleado de la pizzería llega en moto hasta su casa con la pizza en cuestión. El joven paga su compra, le deja una propina al encargado del delivery y luego se dispone a cenar. (Pérez Porto & Gardey, 2018)

2.3.12. MySQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web

que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP). (Rouse, 2015).

2.3.13. API REST

API es un conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas. Para que lo entendamos, el uso de una API es el mecanismo más útil para conectar dos softwares entre sí, de esta manera, podemos garantizar el intercambio de mensajes o datos en formato estándar.

Lo que se pretende con esta herramienta, es simplificar el trabajo de los programadores y desarrolladores puesto que para crearlas o programarlas, no necesitan picar código desde cero, sino hacer solo modificaciones dependiendo el uso que se le va a dar y el tipo de sitio dónde se quiera implementar. Una de las principales funciones de la API es que pueden utilizarse para comunicarse con un sistema operativo que tenga bases de datos o protocolos de comunicaciones. (Ribas Arbós, 2018)

2.3.13.1. Características

- **Protocolo cliente/servidor sin estado:** Cada petición HTTP contiene toda la información necesaria para ejecutarla, por tanto, esto permite que ni cliente ni servidor necesiten recordar ningún estado previo.
- **Cuatro operaciones más importantes:** Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro; POST (crear), GET (leer y consultar), PUT (editar) y DELETE (borrar).
- **Objetos en REST manipulados con URI:** La URI es el identificador único de cada recurso de un sistema REST. Esta, nos facilita el acceso a la información, para poder modificarla o borrarla. También para compartir su ubicación exacta a terceros.

- **Interfaz uniforme:** Para poder realizar una transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto lo que permite es facilitar la existencia de una interfaz uniforme que sistematiza el proceso con la información. (Ribas Arbós, 2018)
- **Sistema de capas:** Su estructura o arquitectura es jerárquica entre sus componentes, y cada una de estas capas, lleva a cabo una funcionalidad dentro del sistema REST.
- **Utilización de hipermedios:** El concepto hipermedio utilizado en los casos de API REST sirve para explicar la capacidad de un interfaz de desarrollo de aplicaciones para proporcionar al cliente y al usuario los enlaces adecuados, y ejecutar acciones concretas sobre los datos. Debemos tener en cuenta que cualquier API debe disponer de hipermedios, puesto que este principio es el que define que cada vez que se hace una petición al servidor y este devuelve una respuesta, parte de la información que contendrá serán los hipervínculos de navegación asociada a otros recursos del cliente. (Ribas Arbós, 2018)

3. PROPUESTA DE SOLUCIÓN

Como propuesta de solución se propone desarrollar una Aplicación Móvil Multiplataforma, Para la Gestión de Pedidos de Comida vía Delivery en los Distritos de Andahuaylas Talavera y San Jerónimo.

3.1. Ingeniería del Proyecto

Para poder desarrollar la aplicación móvil multiplataforma de, gestión de pedidos de comida vía delivery en los distritos de Andahuaylas, San Jerónimo y Talavera constará de gestión del proyecto, metodología de desarrollo y diferentes herramientas tales como software y hardware.

3.2. Base de datos de la aplicación

En la siguiente imagen se muestra el modelo entidad relacion de la base de datos de la aplicación a desarrollar.

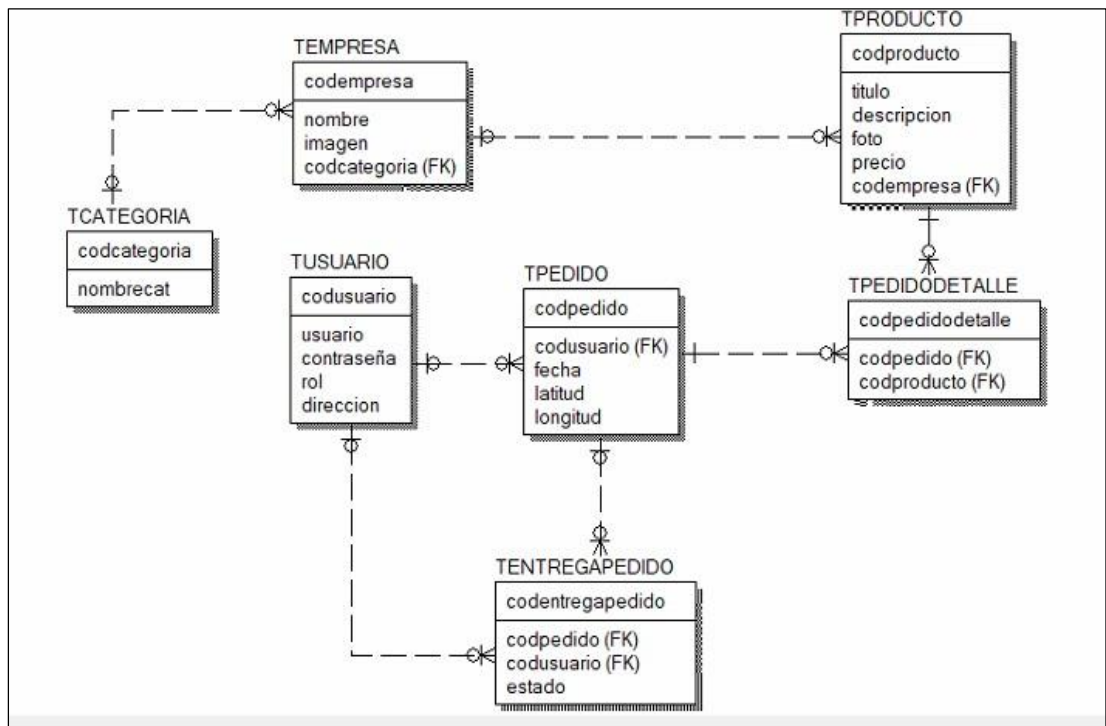


Figura N° 6: Modelo Entidad Relación de la Aplicación

Fuente Elaboración Propia

3.3. Diseño Modular

El desarrollo de la aplicación tendrá los siguientes módulos:

- Módulo Menu

En donde se mostrará las diferentes opciones a las que tendrá acceso el cliente.

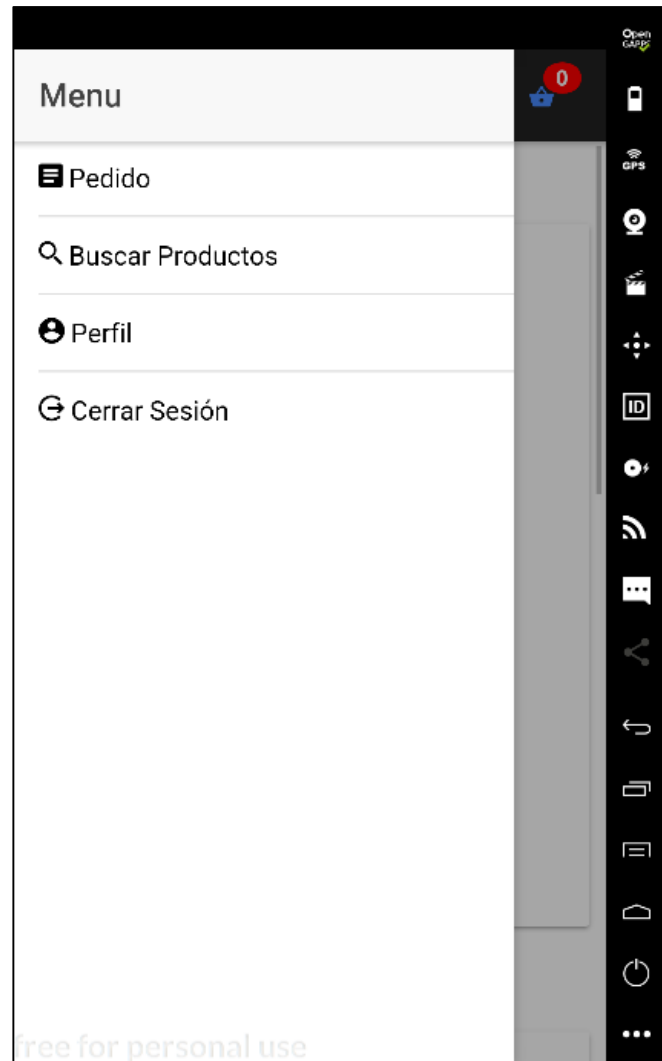


Figura N° 7: Módulo Inicio

Fuente Elaboración Propia

- Módulo Establecimientos

En donde la mostrará los diferentes locales de comida que estarán disponibles para elegir.



Figura N° 8: Módulo Locales

Fuente Elaboración Propia

- Módulo Productos – Local

Este módulo mostrará los diferentes platos o productos que el establecimiento ofrece.

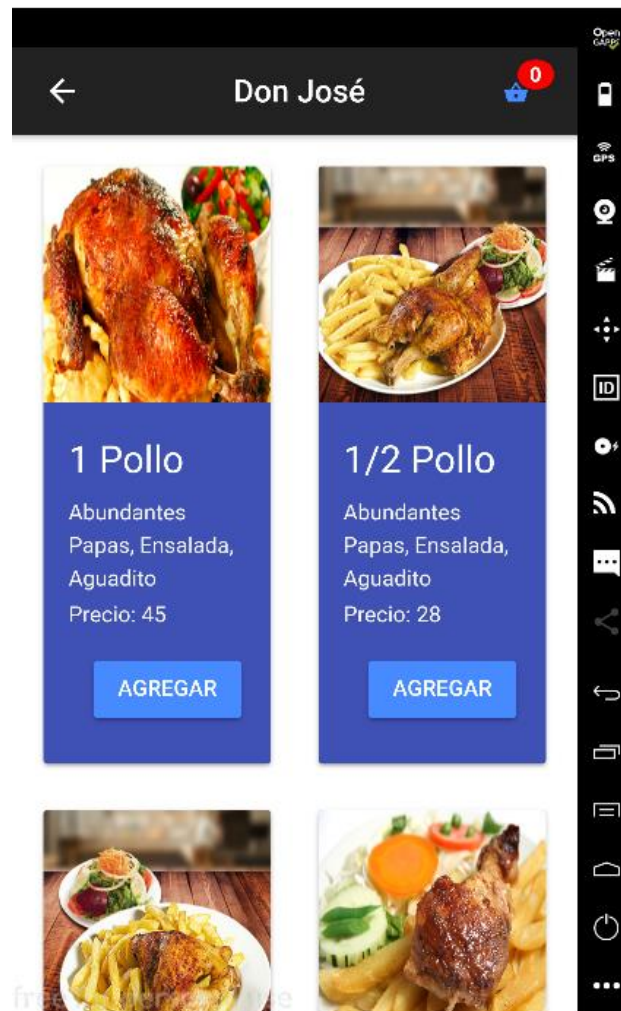


Figura N° 9: Módulo Productos - Local

Fuente Elaboración Propia

- Módulo Login

Este módulo es para la identificación del cliente.



Figura N° 10: Módulo Login

Fuente Elaboración Propia

- Módulo Registro

Este módulo es para que el cliente se registre en la aplicación.

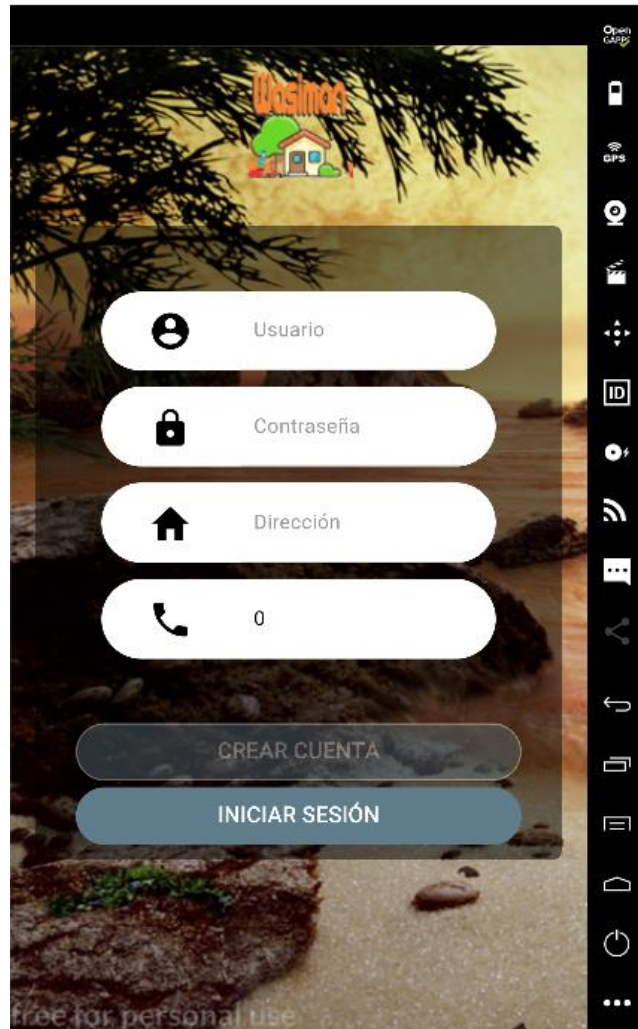
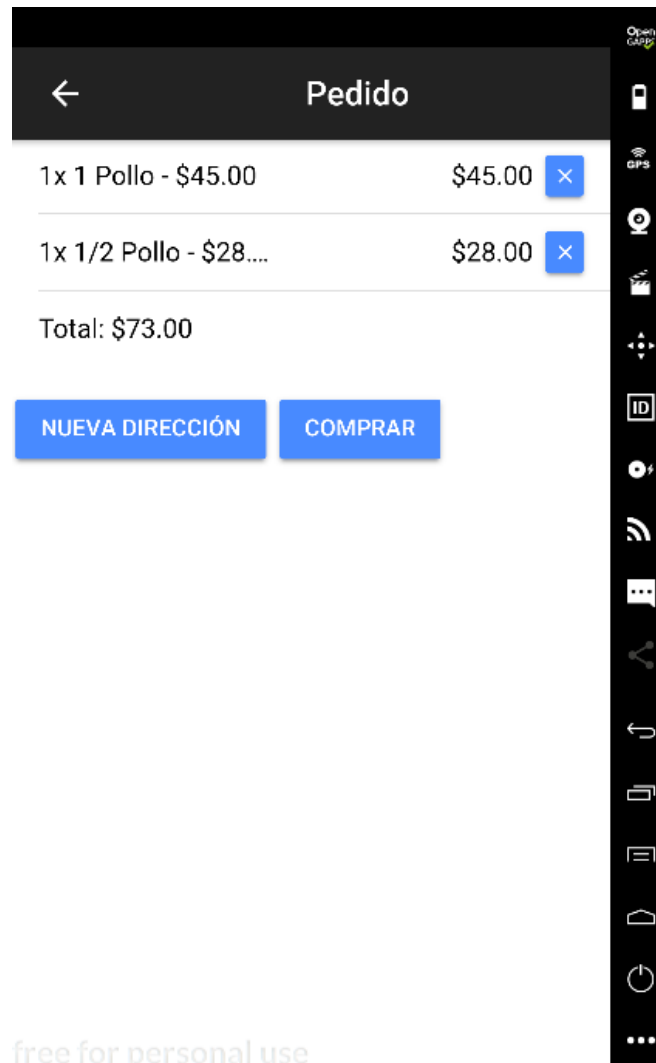


Figura N° 11: Módulo Registro

Fuente Elaboración Propia

- Módulo Pedido

En este módulo el cliente podrá visualizar la lista de productos que serán adquiridos.



free for personal use

Figura N° 12: Módulo Pedido

Fuente Elaboración Propia

- Módulo Comprar

En este módulo el cliente podrá realizar el pago del pedido para concretar el pedido.

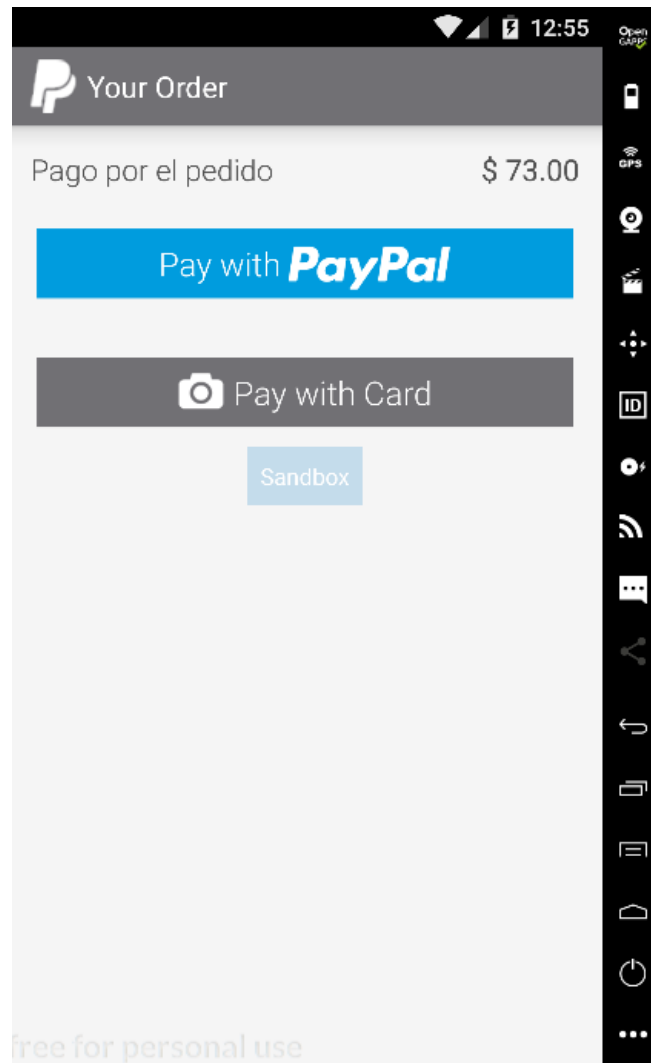


Figura N° 13: Módulo Comprar

Fuente Elaboración Propia

3.4. Uso de metodologías de Programación Mobile-D

Para el desarrollo de este proyecto se utilizó la metodología Mobile-D, esta metodología consta de cinco fases:

- Exploración
- Iniciación
- Producción
- Estabilización
- Pruebas del Sistema

En el siguiente gráfico se observa las diferentes fases que tiene la metodología Mobile-D.



Figura N° 14: Fases de la Metodología Mobile-D

Fuente: Elaboración Propia

3.4.1. Fase de Exploración

En esta fase del proyecto se busca determinar el Plan del Proyecto, nombre que tendrá la aplicación móvil, identificación de usuarios e interesados del proyecto, objetivo general y la plataforma en el cual se desarrollará el proyecto.

3.4.1.1. Establecimiento de Actores

Se deberá identificar a los actores, miembros del equipo que participarán de forma activa en el tiempo que tome desarrollar el proyecto y la aplicación móvil.

- **Jefe del Proyecto:** Responsable de brindar la viabilidad, disponibilidad de todos los recursos que sean necesarios para ejecución del proyecto.
- **Analista de Sistemas:** Responsable del diseño de la infraestructura, interfaz visual de la aplicación móvil. Para ello deberá considerar los requisitos funcionales, el alcance y limitaciones que la aplicación móvil tiene.
- **Programador de Sistemas:** Responsable de la implementación e integración de los módulos que son diseñados por el Analista de Sistemas. De la misma manera será el encargado de realizar los test de prueba de la aplicación móvil.

El desarrollo de este proyecto fue realizado por una sola persona el (Tesisista), cumpliendo estrictamente con los plazos fijados dentro del plan.

3.4.1.2. Estableciendo Usuarios

Se identifican a los diferentes tipos de usuario que harán uso de la aplicación móvil.

- **Usuarios Directos:** Clientes de los diferentes establecimientos que se dedican a la venta de comida en los distritos de Andahuaylas, Talavera y San Jerónimo.
- **Usuarios Indirectos:** Diferentes personas que se dedican al desarrollo de aplicaciones móviles utilizan frameworks que son utilizados en este proyecto.

3.4.1.3. Información Clave del Proyecto

Se definió lo siguiente:

- Nombre comercial de la aplicación móvil.
- Objetivo general.
- Plataformas en las cuales funcionará el proyecto.

Tabla N° 2: Información del Proyecto

NOMBRE DE LA APLICACIÓN	Wasiman
OBJETIVO GENERAL	Desarrollar una aplicación móvil, que permita la gestión de comida vía delivery a los clientes de los diferentes locales de comida en los distritos de Andahuaylas, Talavera y San Jerónimo.
PLATAFORMA	Android, los, Windows Phone

Fuente: Elaboración Propia

3.4.1.4. Principales Requerimientos Funcionales

- La estructura de la aplicación móvil será escalable y robusta.
- La aplicación móvil será utilizada solo por usuarios autenticados.
- La aplicación móvil estará disponible en las plataformas de Android, iOS y Windows Phone.

- La aplicación móvil mostrará los diferentes platillos que ofrecen los establecimientos de comida en los distritos de Andahuaylas, Talavera y San Jerónimo.

3.4.1.5. Principales Requerimientos NO Funcionales

- El framework a utilizar para el desarrollo de esta aplicación será IONIC 3, MySQL server, y plugins de Cordova.
- la aplicación móvil deberá ser intuitiva para su uso.
- La aplicación móvil tendrá interfaces gráficas bien estructuradas.
- La aplicación móvil tendrá un diseño responsive.
- La aplicación móvil no debe ocupar más de 35MB en el dispositivo móvil instalado.

3.4.2. Fase de Iniciación

Se define las herramientas de desarrollo, hardware para hacer las pruebas y diagramas de caso de uso.

3.4.2.1. Materiales de Desarrollo

- IONIC 3
- AngularJS
- MySQL
- Visual Studio Code
- Apache Cordova
- Toad for MySQL

3.4.2.2. Hardware

- 01 laptop Toshiba Satellite Core i7, Procesador 2.40GHz, Sistema Operativo Windows 10.

3.4.2.3. Herramientas de Prueba

- 01 Smart Phone Samsung Galaxy J5 Prime
- 01 Smart Phone iPhone 8
- 01 laptop Toshiba Satellite Core i7, Sistema operativo Windows 10.

3.4.2.4. Diagramas de casos de uso de la aplicación Móvil

- **Vista Login:** Muestra un formulario de acceso a la aplicación móvil, tiene las siguientes acciones login, registrarse, recuperar contraseña.

Tabla N° 3: Caso de Uso Iniciar Aplicación

CASO DE USO	Iniciar aplicación
ACTORES	Usuario
PROPÓSITO	Dar acceso a la vista principal de la aplicación móvil.
RESUMEN	El usuario deberá ingresar los datos que solicita la aplicación que son (usuario y contraseña).
TIPO	Primario
PRE-CONDICIONES	Abrir la aplicación

Fuente: Elaboración Propia

- **Crear Nueva Cuenta:** Muestra un formulario en donde el usuario deberá registrar sus datos reales para poder acceder a la aplicación móvil.

Tabla N° 4: Caso de Uso Crear Nueva Cuenta

CASO DE USO	Crear Nueva Cuenta
ACTORES	Usuario
PROPÓSITO	Registrar datos del usuario, para luego registrarlo en la base de datos de la aplicación y el usuario pueda hacer su uso.
RESUMEN	El usuario deberá ingresar verdaderos para que pueda hacer uso de la aplicación.
TIPO	Primario
PRE-CONDICIONES	Haber seleccionado la opción de “Crear nueva Cuenta”

Fuente: Elaboración Propia

- **Recuperar Contraseña:** Muestra un formulario para recuperar la contraseña de acceso a la aplicación móvil.

Tabla N° 5: Caso de Uso Recuperar Contraseña

CASO DE USO	Recuperar Contraseña
ACTORES	Usuario
PROPÓSITO	Posibilidad de recuperar la contraseña del usuario.
RESUMEN	Mostrar al usuario la pregunta secreta que respondió al momento de registrarse, si lo responde correctamente le dará la opción de establecer una nueva contraseña.
TIPO	Primario

PRE-CONDICIONES	Haber seleccionado el botón “Recuperar Contraseña” del menú Login.
------------------------	--

Fuente: Elaboración Propia.

- **Menú Principal:** Muestra los diferentes establecimientos de comida agrupadas por categorías en donde el usuario podrá seleccionar el establecimiento que le convenga.

Tabla N° 6: Caso de Uso Menú Principal

CASO DE USO	Menú Principal
ACTORES	Usuario
PROPÓSITO	Muestra los establecimientos registrados en la aplicación por categorías.
RESUMEN	El usuario podrá visualizar los diferentes establecimientos de comida que están registrados en la aplicación.
TIPO	Primario
PRE-CONDICIONES	Haberse logueado de manera satisfactoria en la aplicación.

Fuente: Elaboración Propia

- **Seleccionar Establecimiento:** Una vez el usuario haya seleccionado el establecimiento la aplicación móvil mostrará los diferentes platillos que ofrece el establecimiento seleccionado.

Tabla N° 7: Caso de Uso Seleccionar Establecimiento

CASO DE USO	Seleccionar establecimiento
ACTORES	Usuario
PROPÓSITO	Mostrar platillos de cada establecimiento.

RESUMEN	Mostrar al usuario los diferentes platillos que ofrece el establecimiento seleccionado.
TIPO	Primario
PRE-CONDICIONES	Haber seleccionado el establecimiento de su preferencia.

Fuente: Elaboración Propia.

- **Buscar Producto:** Ese módulo permite al usuario la búsqueda de productos que desee, sin la necesidad de indagar por los establecimientos que existe en la aplicación móvil.

Tabla N° 8: Caso de Uso Buscar Producto

CASO DE USO	Buscar Producto
ACTORES	Usuario
PROPÓSITO	Ayudar a buscar un producto específico en todos los establecimientos al usuario.
RESUMEN	Mostrar al usuario el platillo en específico que buscó.
TIPO	Primario
PRE-CONDICIONES	Haber seleccionado "Buscar Producto".

Fuente: Elaboración Propia

- **Agregar Producto:** Esta opción permitirá al usuario agregar los productos que desee realizar el pedido al carrito de compras.

Tabla N° 9: Caso de Uso Agregar Producto

CASO DE USO	Agregar Producto
ACTORES	Usuario
PROPÓSITO	Agregar el platillo al carrito de Compras

RESUMEN	Agregar los platillos seleccionados por el usuario al carrito de compras.
TIPO	Primario
PRE-CONDICIONES	Seleccionar el botón “Agregar” de los platillos.

Fuente: Elaboración Propia.

- **Listar Pedido:** Muestra al usuario todos los productos y la suma total del precio de los productos que fueron agregados al carrito de compras.

Tabla N° 10: Caso de Uso Listar Pedido

CASO DE USO	Listar Pedido
ACTORES	Usuario
PROPÓSITO	Mostrar al usuario la lista de platillos que tiene en su carrito de compras.
RESUMEN	El usuario podrá visualizar los precios unitarios y el total de todos los platillos que haya agregado al carrito de compras.
TIPO	Primario
PRE-CONDICIONES	Seleccionar la opción “Listar Pedido” del menú Principal.

Fuente: Elaboración Propia.

- **Carrito de Compras:** Muestra al usuario todos los productos que están listos para generar el pedido.

Tabla N° 11: Caso de Uso Carrito de Compras

CASO DE USO	Carrito de Compras
ACTORES	Usuario

PROPÓSITO	Listar todos los productos que se seleccionaron.
RESUMEN	Esta opción mostrará al usuario la lista de todos los pedidos que tiene en su carrito de compras
TIPO	Primario
PRE-CONDICIONES	Seleccionar el icono “Carrito de Compras” que está ubicado en la parte superior derecha de la aplicación.

Fuente: Elaboración Propia.

- **Cambiar Dirección:** Permite al usuario modificar la dirección que consignó al momento de registrarse en la aplicación móvil.

Tabla N° 12: Caso de Uso Cambiar Dirección

CASO DE USO	Cambiar Dirección
ACTORES	Usuario
PROPÓSITO	Posibilidad de cambiar la dirección de entrega del pedido.
RESUMEN	Dar la posibilidad de cambiar la dirección de entrega del pedido por otra dirección que no sea la que se consignó al momento de registrarse.
TIPO	Primario
PRE-CONDICIONES	Seleccionar la opción “Cambiar Dirección” en la sección de Generar Pedido.

Fuente: Elaboración Propia.

- **Generar Pedido:** Esta opción permite al usuario concretar el pedido de los productos que se encuentran en el carrito de compras, para posteriormente realizar el pago.

Tabla N° 13: Caso de Uso Generar Pedido

CASO DE USO	Generar Pedido
ACTORES	Usuario
PROPÓSITO	Contrastar el pedido del usuario.
RESUMEN	Esta opción servirá para que el usuario confirme el pedido para luego proceder a realizar el pago correspondiente.
TIPO	Primario
PRE-CONDICIONES	Seleccionar la opción “Generar Pedido” después de haber seleccionado el carrito de compras.

Fuente: Elaboración Propia.

- **Comprar:** Esta opción permite al usuario realizar el pago correspondiente para generar el pedido.

Tabla N° 14: Caso de Uso Comprar

CASO DE USO	Comprar
ACTORES	Usuario
PROPÓSITO	Realizar el pago.
RESUMEN	Esta opción servirá para que el usuario realice el pago por el pedido de los productos que haya seleccionado.
TIPO	Primario

PRE-CONDICIONES	Seleccionar botón “Comprar” después de haber seleccionado el carrito de compras.
------------------------	--

Fuente: Elaboración Propia.

3.4.3. Fase de Producción

En la fase de se diseña la aplicación móvil de acuerdo a los diferentes casos de uso que se desarrolló en la fase de iniciación, en donde se determinó los principales requerimientos del proyecto, de esta manera poder lograr la funcionalidad que tendrá la aplicación móvil.

En la implementación de la aplicación móvil se llevó acabo utilizando un ciclo de desarrollo iterativo, esto determina que cada módulo desarrollado fue probado.

En cada una de las iteraciones las actividades realizadas fueron:

- ✓ Análisis de requerimientos de la iteración
- ✓ Desarrollo de la iteración
- ✓ Integración del nuevo módulo desarrollado al esqueleto de la aplicación.
- ✓ Retroalimentación

3.4.3.1. Diseño de la Aplicación móvil

Tomando en cuenta los análisis realizados en la fase de iniciación, en donde se definió los requerimientos de la aplicación móvil. En la siguiente tabla se define la estructura de la aplicación móvil en términos generales.

Tabla N° 15: Estructura de la Aplicación Móvil

ACCESO A LA APLICACIÓN	Estarán disponible las opciones de inicio de sesión, recuperar contraseña, crear nueva cuenta.
PÁGINA PRINCIPAL	Mostrará los diferentes establecimientos de comida seleccionadas por categorías.
CONFIGURACIÓN DE PERFIL	Permitirá al usuario a modificar los datos de cuenta.
ETABLECIMIENTO	Mostrará los diferentes productos que ofrece el establecimiento seleccionado por el usuario.
BASE DE DATOS	El proyecto tendrá una base de datos alojado en un servidor.

Fuente: Elaboración Propia

3.4.3.1. Dependencias de Paquetes de la Aplicación móvil

Para usar las diferentes funcionalidades de Ionic, debemos importar diferentes paquetes, estos paquetes son de uso general. En el siguiente gráfico se muestra las dependencias utilizadas para este proyecto.

```
"dependencies": {
  "@angular/animations": "5.2.11",
  "@angular/cli": "^7.3.5",
  "@angular/common": "5.2.11",
  "@angular/compiler": "5.2.11",
  "@angular/compiler-cli": "5.2.11",
  "@angular/core": "5.2.11",
  "@angular/forms": "5.2.11",
  "@angular/http": "5.2.11",
  "@angular/platform-browser": "5.2.11",
  "@angular/platform-browser-dynamic": "5.2.11",
  "@ionic-native/core": "^4.18.0",
```

```

"@ionic-native/google-maps": "^4.8.2",
"@ionic-native/paypal": "^4.20.0",
"@ionic-native/splash-screen": "~4.18.0",
"@ionic-native/status-bar": "~4.18.0",
"@ionic/pro": "2.0.4",
"@ionic/storage": "2.2.0",
"angular-route": "^1.7.7",
"card.io.cordova.mobilesdk": "2.1.0",
"com.paypal.cordova.mobilesdk": "3.5.0",
"cordova-android": "7.1.4",
"cordova-plugin-device": "^2.0.2",
"cordova-plugin-googlemaps": "2.3.4",
"cordova-plugin-ionic-keyboard": "^2.1.3",
"cordova-plugin-ionic-webview": "^3.1.2",
"cordova-plugin-splashscreen": "^5.0.2",
"cordova-plugin-statusbar": "^2.4.2",
"cordova-plugin-whitelist": "^1.3.3",
"ionic-angular": "3.9.3",
"ionicons": "3.0.0",
"rxjs": "^5.5.11",
"sw-toolbox": "3.6.0",
"zone.js": "0.8.29"
},
"devDependencies": {
  "@ionic/app-scripts": "3.2.1",
  "typescript": "~2.6.2"
},

```

Figura N° 15: Dependencias de la Aplicación

Fuente: Elaboración Propia

3.4.3.2. Plugins Necesarios para la Aplicación Móvil-Cordova

Cada plugin se instala en la aplicación móvil de acuerdo a una necesidad y son de uso pleno de los dispositivos móviles de acuerdo a la plataforma en que se compilará, estos plugins permiten el acceso de la aplicación móvil a la cámara, GPS, micrófono, etc.

En el siguiente gráfico se muestra los plugins utilizados para el desarrollo de la aplicación móvil.

```

"cordova": {
  "plugins": {
    "cordova-plugin-whitelist": {},
    "cordova-plugin-statusbar": {},
    "cordova-plugin-device": {},
    "cordova-plugin-splashscreen": {},
    "cordova-plugin-ionic-webview": {
      "ANDROID_SUPPORT_ANNOTATIONS_VERSION": "27.+"
    },
    "cordova-plugin-ionic-keyboard": {},
    "com.paypal.cordova.mobilesdk": {},
    "cordova-plugin-googlemaps": {
      "API_KEY_FOR_ANDROID":
"AIzaSyCfS038i4L0zEqPztkiP0fPkZiHzMqyv_g",
      "API_KEY_FOR_IOS": "YOUR_IOS_API_KEY_IS_HERE",
      "PLAY_SERVICES_VERSION": "15.0.1",
      "ANDROID_SUPPORT_V4_VERSION": "26.1.0"
    }
  },
  "platforms": [
    "android"
  ]
}

```

Figura N° 16: Plugins Necesarios para la Aplicación Móvil-Cordova

Fuente: Elaboración Propia

3.4.3.3. Configuración Inicial del Proyecto

En primer lugar, para crear el proyecto se debe teclear el siguiente comando en la terminal de comandos:

\$ ionic start miApp blank

Este comando permite la creación de toda la estructura del proyecto, junto a esto también se creará el archivo Package.json, que contiene todas las configuraciones de NodeJS.

En el archivo “app.module.ts” se deberá importar todas las dependencias y páginas que se usarán en el desarrollo del proyecto. En el siguiente gráfico se muestra la configuración del archivo app.module.ts.

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { ErrorHandler, NgModule, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
3 import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
4 import { IonicStorageModule } from '@ionic/storage';
5 import { MyApp } from './app.component';
6 import { HomePage } from '../pages/home/home';
7 import { LoginPage } from '../pages/login/login';
8 import { CartPage } from '../pages/cart/cart';
9 import { MapPage } from '../pages/map/map';
10 import { SearchPage } from '../pages/search/search';
11 import { RegisterPage } from '../pages/register/register';
12 import { ProductoPage } from '../pages/producto/producto';
13 import { HttpClientModule } from '@angular/http';
14 import { StatusBar } from '@ionic-native/status-bar';
15 import { SplashScreen } from '@ionic-native/splash-screen';
16 import { UserStorageService } from '../services/user-storage.service';
17 import { CartService } from '../services/cart.service';
18 import { UserService } from '../services/services-rest/user.service';
19 import { PedidoService } from '../services/services-rest/pedido.service';
20 import { EntregaPedidoService } from '../services/services-rest/entregaPedido.service';
21 import { PedidoDetalleService } from '../services/services-rest/pedidoDetalle.service';
22 import { ProductoService } from '../services/services-rest/producto.service';
23 import { EmpresaService } from '../services/services-rest/empresa.service';
24 import { HttpService } from '../core/http.service';
25 import { CartcountComponent } from '../components/cartcount/cartcount';
26 import { PayPal } from '@ionic-native/paypal';
27 import { GoogleMaps } from '@ionic-native/google-maps';
28 import { MapService } from '../services/map.service';
29 import { ToastService } from '../services/toast.service';
30 import { EntregaPage } from '../pages/entrega/entrega';
31 import { AsignarPedidoPage } from '../pages/asignar-pedido/asignar-pedido';
32 @NgModule({
33   declarations: [
```

Figura N° 17: Configuración de app.module.ts

Fuente: Elaboración Propia

3.4.3.4. Estructura de los directorios y archivos del proyecto

Al momento de ejecutar el comando **\$ ionic start miApp blank**, este se encarga de crear la estructura de directorios y archivos necesarios de manera automática para empezar con la codificación de la aplicación móvil.

En el siguiente gráfico se muestra la estructura de directorios y archivos del proyecto.

- **Directorio PEDIDO:**

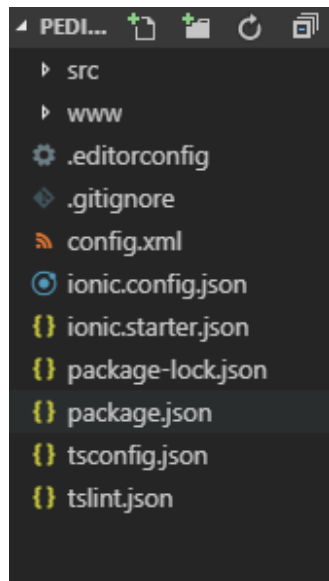


Figura N° 18: Estructura del Directorio PEDIDO

Fuente: Elaboración Propia

Tabla N° 16: Directorios del Proyecto PEDIDOS

src	En este directorio es muy importante, porque en ella estarán alojadas todos los archivos de configuración y de programación que forman el proyecto y la aplicación.
www	Contiene archivos javascript en una carpeta build e imágenes estáticas para el icono de la aplicación y el SplashScreen, por default.
package.json	Es un archivo json que contiene información general del proyecto, las dependencias y paquetes y los plugins instalados. Es un archivo muy importante que sirve para migrar el proyecto e instalar nuevamente de ser necesario.
.editorconfig, .gitignore,	Todo estos archivos son creados y usados por el framework IONIC 3.

config.xml, ionic.config.json, ionic.starter.json, package- lock.json, tsconfig.json, tslint.json	
---	--

- **Directorio src:**

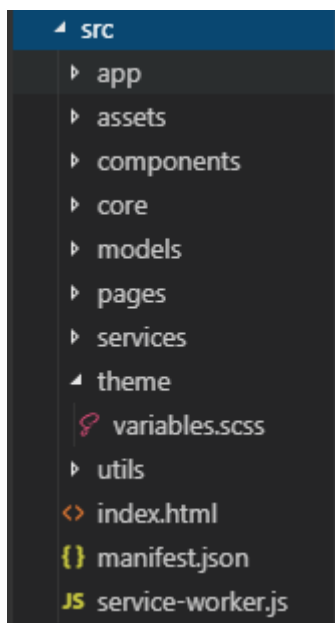


Figura N° 19: Directorio src

Fuente: Elaboración Propia

- **Tipos de Archivos Utilizados en el Proyecto**

Tabla N° 17: Tipos de archivos utilizados en el Proyecto

Archivos “.ts” (TypeScript)	TypeScript es un lenguaje de programación libre y de código abierto desarrollado por Microsoft, es un super conjunto de JavaScript con herramientas
-----------------------------	---

	de programación orientada a objetos y añade tipado estático.
Archivos “.HTML”	En este proyecto es el típico archivo html pero con etiqueta ion de ionic y uso de directivas y funcionalidades de angular.
Archivos Sass “.scss”	SASS es una extensión de CSS escrita en RUBY que añade características muy potentes y elegantes a este lenguaje de estilos (CSS). Permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchos más.

Fuente: Elaboración Propia

- **Directorio app:**

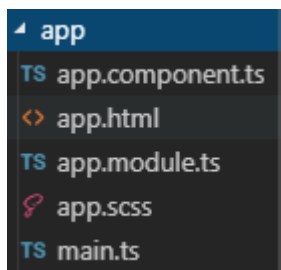


Figura N° 20: Directorio app

Fuente: Elaboración Propia

Tabla N° 18: Directorio app del Proyecto

app.component.ts	Redirige la página, para que cargue la página de inicio de sesión como página raíz.
app.module.ts	Este archivo se instancia los paquetes y dependencia utilizados en este proyecto.
app.scss, main.ts, app.html	Estos archivos no se modificarán se dejarán tal como está al momento

	de ejecutar el comando \$ionic start miApp blank.
--	---

Fuente: Elaboración Propia

- **Directorio assets:**

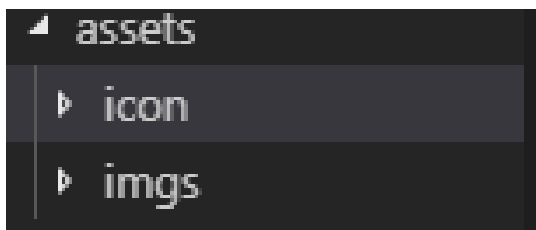


Figura N° 21: Directorio assets

Fuente: Elaboración Propia

Tabla N° 19: Directorio assets

assets	Es un directorio en donde irán las iconos e imágenes estáticas de la aplicación móvil.
--------	--

Fuente: Elaboración Propia

- **Directorio components:**

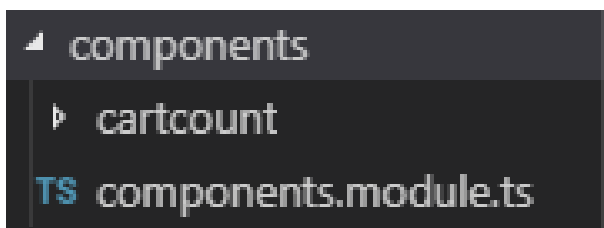


Figura N° 22: Directorio components

Fuente: Elaboración Propia

Tabla N° 20: Directorio components

components	Bloques de código que se puede utilizar en cualquier página dentro del proyecto.
------------	--

Fuente: Elaboración Propia

- **Directorio core:**

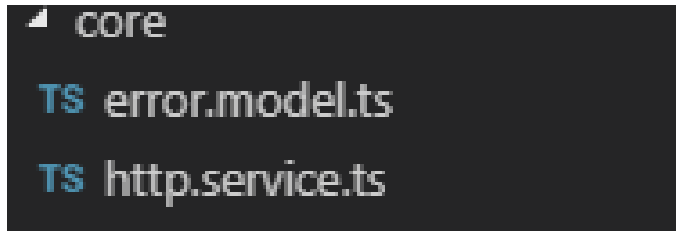


Figura N° 23: Directorio core

Fuente: Elaboración Propia

Tabla N° 21: Directorio core

core	En este directorio se crearán las clases principales que se podrán reutilizar en otros proyectos.
------	---

Fuente: Elaboración Propia

- **Directorio models:**

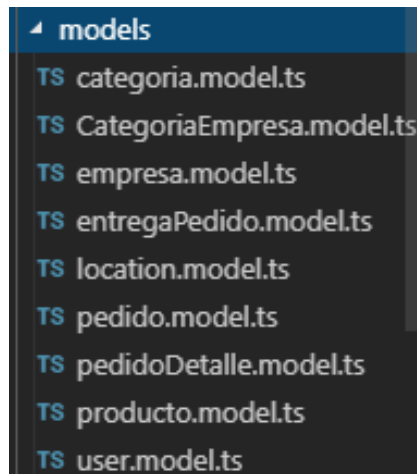


Figura N° 24: Directorio models

Fuente: Elaboración Propia

Tabla N° 22: Directorio models

Model.ts	Realizar operación de get, post
----------	---------------------------------

Fuente: Elaboración Propia

- **Directorio pages:**

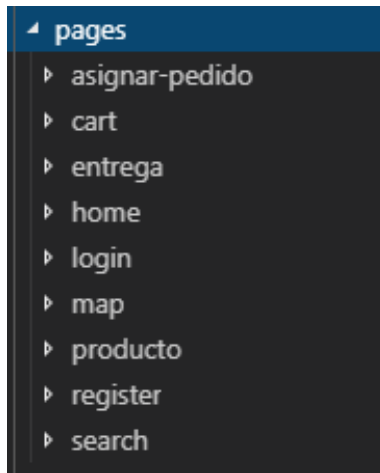


Figura N° 25: Directorio pages

Fuente: Elaboración Propia

Tabla N° 23: Directorio pages

asignar-pedido	asignar-pedido.html	Diseño para la asignación de pedido de parte del administrador al trabajador.
	asignar-pedido.scss	Estilos para la visualización de la información.
	asignar-pedido.ts	Definición de variables, importación de servicios de enrutamiento a las páginas definidas y método de cerrar sesión.
cart	cart.html	Diseño del carrito de compras.
	cart.scss	Estilos al carrito de compras.
	cart.ts	Utilización de servicios.
entrega	entrega.html	Diseño de la interfaz de entregar pedido.

	entrega.scss	Personalización a la interfaz entregar pedido.
	entrega.ts	Importar diferentes servicios, para hacer funcional los diferentes elementos de la interfaz.
home	home.html	Arquitectura de los diferentes elementos que irán en la ventana.
	home.scss	Estilos css a los elementos del archivo home.html
	home.ts	Servicios que permitan el acceso a menú principal de la aplicación.
login	login.html	Diseño del formulario login.
	login.scss	Estilos para el formulario login.
	login.ts	Importación de servicios, acceso a otros formularios, revisión de estado de conexión a internet, autenticación de usuario por credenciales e inicio de sesión.
map	map.html	Definición de las dimensiones del mapa.
	map.scss	Estilos para el mapa.
	map.ts	Importación de los servicios de GoogleMaps.
producto	producto.html	Diseño de la interfaz de los productos que ofrecen los establecimientos.
	producto.scss	Estilos de las interfaces de los productos.

	producto.ts	Importación de servicios para realizar la carga de productos.
register	register.html	Diseño del formulario para poder crear una nueva cuenta en la aplicación móvil.
	register.scss	Estilos al formulario crear nueva cuenta.
	register.ts	Importación de servicios y módulos para crear nueva cuenta.
search	search.html	Diseño del buscador de productos.
	search.scss	Estilos al buscador de productos.
	search.ts	Importación de servicios para que el buscador sea funcional.

Fuente: Elaboración Propia

- **Directorio services:**

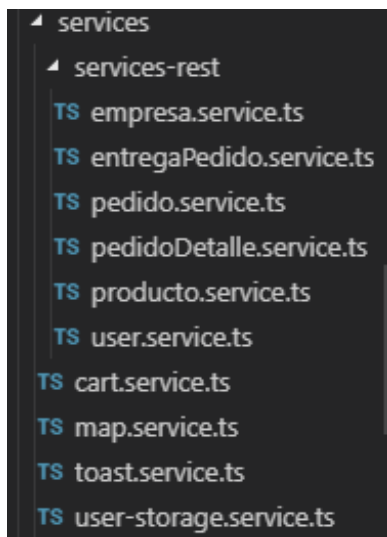


Figura N° 26: Directorio services

Fuente: Elaboración Propia

Tabla N° 24: Directorio services

Services-rest	empresa.service.ts	Llama a la API empresa en la cual se obtiene información de la tabla empresa que se encuentra en la base de datos, además aquí se escribirán funciones para agregar empresas.
	entregaPedido.service.ts	Llama a la API entregaPedido en la cual se obtiene información de la tabla entregaPedido que se encuentra en la base de datos, además aquí se escribirán funciones para la entrega de pedidos.
	pedido.service.ts	Llama a la API pedido en la cual se obtiene información de la tabla pedido que se encuentra en la base de datos, además aquí se escribirán funciones para generar pedido.
	pedidoDetalle.service.ts	Llama a la API pedidoDetalle en la cual se obtiene información de la tabla pedidoDetalle que se encuentra en la base de datos.
	producto.service.ts	Llama a la API producto en la cual se obtiene información de la tabla producto que se

		encuentra en la base de datos, de la misma manera se escribirán funciones para agregar productos..
	user.service.ts	Llama a la API usuarios en la cual se obtiene información de la tabla user que se encuentra en la base de datos.
services	cart.service.ts	Es una clase en donde se encuentra la api que interconectará a la aplicación móvil con la base de datos.
	map.service.ts	Es una clase en donde se encuentra la api que interconectará a la aplicación móvil con los servicios de google maps.
	toast.service.ts	Es una clase en donde se encuentra la api que interconectará a la aplicación móvil con los servicios de toast.

Fuente: Elaboración Propia

3.4.4. Fase de Estabilización

En esta fase se procedió a integrar las funcionalidades implementadas en los diferentes módulos y de presentarse algún error de código u otros se realizarán las correcciones correspondientes.

El propósito de esta fase es la aseguración de la calidad en la implementación del proyecto.

3.4.4.1. Integración del proyecto con la base de datos

La aplicación móvil hace el uso de una API REST, el cual permite interactuar con una base de datos que se encuentra en un servidor en la red.

3.4.4.2. Integración de los módulos de Acceso a la aplicación

Debemos establecer como página principal login en el archivo app.component.ts

```
@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  @ViewChild(Nav) nav: Nav;

  rootPage: any = 'login';
```

Figura N° 27: Página Raiz

Fuente: Elaboración Propia

- El formulario login está diseñado con las siguientes etiquetas:

```
<ion-content class="login-content" padding>
  <ion-row class="logo-row">
    <ion-col></ion-col>
    <ion-col class="logo">
    </ion-col>
    <ion-col></ion-col>
  </ion-row>
  <div class="login-box">
    <form #loginForm="ngForm">
      <ion-row>
        <ion-col>
          <ion-list inset>
            <ion-item>
              <ion-icon name="contact" item-start></ion-icon>
              <ion-input type="text" placeholder="Usuario"
name="usuario" [(ngModel)]="credentials.usuario"
required></ion-input>
            </ion-item>
```



```

        <ion-item>
            <ion-icon name="lock" item-start></ion-icon>
            <ion-input type="password"
placeholder="Contraseña" name="contrasena"
[(ngModel)]="credentials.contrasena" required></ion-input>
        </ion-item>
    </ion-list>
</ion-col>
</ion-row>
<ion-row>
    <ion-col class="signup-col">
        <button ion-button class="submit-btn" full
(click)="login()" [disabled]="!loginForm.form.valid">Iniciar
sesión</button>
        <button ion-button class="register-btn" block clear
(click)="showRegister()">Crear nueva cuenta </button>
    </ion-col>
</ion-row>
</form>
</div>
</ion-content>

```

Figura N° 28: login.html

Fuente: Elaboración Propia

- Contenido del archivo login.ts:

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { User } from '../models/user.model';
import { UserService } from '../services/services-
rest/user.service';
import { UserStorageService } from '../services/user-
storage.service';
import { HomePage } from '../pages/home/home';
import { RegisterPage } from '../pages/register/register';
import { AsignarPedidoPage } from '../asignar-pedido/asignar-
pedido';
import { EntregaPage } from '../entrega/entrega';
@IonicPage()
@Component({
    selector: 'page-login',
    templateUrl: 'login.html',
})
export class LoginPage {
    credentials: User = { usuario: '', contrasena: '' };

```

```

    constructor(public navCtrl: NavController, public
navParams: NavParams, public userService: UserService, public
userService: UserService) {
    }

    ionViewDidLoad() {
        console.log('ionViewDidLoad LoginPage');
    }
    login() {
        this.userService.logIn(this.credentials).subscribe(
            (data) => {
                console.log(data);

this.userService.setIdUser(data['users'].codusuario);
                this.userService.setRol(data['users'].rol)
                if (data['users'].rol == "admin") {
                    this.navCtrl.setRoot(AsignarPedidoPage);
                } else if (data['users'].rol == "trabajador") {
                    this.navCtrl.setRoot(EntregaPage);
                } else if (data['users'].rol == "usuario") {
                    this.navCtrl.setRoot(HomePage);
                }

            },
            (error) => {
                console.log(error);
            }
        );
    }
    showRegister() {
        this.navCtrl.push(RegisterPage);
    }
}

```

Figura N° 29:login.ts

Fuente: Elaboración Propia

- Contenido del archivo asignar-pedido.html

```

<ion-header>
  <ion-navbar color="dark">
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Asignar pedido</ion-title>
  </ion-navbar>
</ion-header>

```

```

<ion-content>
  <ion-list>
    <ion-list-header>Pedidos Asignados</ion-list-header>
    <ion-row *ngFor="let item of entregaPedidos; let i =
index">
      <ion-col>
        <ion-item>
          <ion-icon name="appstore" item-start</ion-
icon>
          <h2>Pedido #{{item.codpedido.codpedido}}</h2>
          <p>Trabajador: {{item.codusuario.usuario}}</p>
          <p>Cliente:
{{item.codpedido.codusuario.usuario}}</p>

          </ion-item>
        </ion-col>
      </ion-row>
    </ion-list>
    <ion-list>
      <ion-list-header>Pedidos pendientes</ion-list-header>
      <ion-row *ngFor="let item of pedidos; let i = index">
        <ion-col col-8 (click)="showDetail(item)">
          <ion-item>
            <h2>Pedido #{{i + 1}}</h2>
            <p>Cliente: {{item.codusuario.usuario}}</p>

            </ion-item>
          </ion-col>
          <ion-col col-4>
            <ion-item>
              <ion-select class="myCustomSelect" #C
(ionChange)="updateEstado(C.value,item.codpedido)">
                <ion-option *ngFor="let item of usuarios"
[value]="item.codusuario" >{{item.usuario}}</ion-option>
                <ion-option value="-1" [selected]="true"
>Seleccionar</ion-option>

              </ion-select>
            </ion-item>
          </ion-col>
        </ion-row>
      </ion-list>
    </ion-content>

```

Figura N° 30: asignar-pedido.html

Fuente: Elaboración Propia

- Contenido del archivo asignar-pedido.ts:

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { PedidoService } from '../../services/services-rest/pedido.service';
import { UserService } from '../../services/services-rest/user.service';
import { EntregaPedidoService } from
'../../services/services-rest/entregaPedido.service';
import { ToastService } from '../../services/toast.service';

@IonicPage()
@Component({
  selector: 'page-asignar-pedido',
  templateUrl: 'asignar-pedido.html',
})
export class AsignarPedidoPage {
  pedidos: any;
  usuarios: any;
  entregaPedidos: any;
  constructor(public navCtrl: NavController, public
navParams: NavParams, public pedidoService: PedidoService,
public userService: UserService, public entregaPedidoService:
EntregaPedidoService, public toastService: ToastService) {
  this.pedidoService.findAll().subscribe(
    (data) => {
      this.pedidos = data['pedidos'];
      console.log(this.pedidos);
    }
  )
  this.findAllEntregaPedidos();
  this.findByRol();
}
findAllEntregaPedidos() {
  this.entregaPedidoService.findAll().subscribe(
    (data) => {
      this.entregaPedidos = data['entregaPedidos'];
    }
  )
}
findByRol() {
  this.userService.findByRol('trabajador').subscribe(
    (data) => {
      this.usuarios = data['users'];
    }
  )
}
```



```

    Total: <span slot="end">{{ total |
currency:'USD': 'symbol' }}</span>
  </ion-item>
  <ion-item *ngIf="latLng">
    Nueva dirección: <span slot="end" >{{
latLng.direccion}}</span>
  </ion-item>
</ion-list>

  <button ion-button color="primary"
(click)="addDireccion()">nueva dirección</button>
  <button ion-button color="primary"
(click)="pedir()">Comprar</button>
</ion-content>

```

Figura N° 32: cart.html

Fuente: Elaboración Propia

- Contenido del archivo cart.ts:

```

import { Component, OnInit } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { CartService } from '../services/cart.service';
import { PedidoService } from '../services/services-
rest/pedido.service';
import { PedidoDetalleService } from '../services/services-
rest/pedidoDetalle.service';
import { Pedido } from '../models/pedido.model';
import { Producto } from '../models/producto.model';
import { PedidoDetalle } from
'../models/pedidoDetalle.model';
import { UserStorageService } from '../services/user-
storage.service';
import { PayPal, PayPalPayment, PayPalConfiguration } from
 '@ionic-native/paypal';
import { ToastController } from 'ionic-angular';
import { ToastService } from '../services/toast.service';
import { MapPage } from '../map/map';
@IonicPage()
@Component({
  selector: 'page-cart',
  templateUrl: 'cart.html',
})
export class CartPage implements OnInit {
  productos = [];
  total = 0;

```

```

codUser="";
latLng;
constructor(private toastService:ToastService,private
paypal: PayPal,public navCtrl: NavController, public
navParams: NavParams, private cartService:CartService,
private pedidoService:PedidoService,private
userService:UserService,private
pedidoDetalleService:PedidoDetalleService) {
  this.userService.getIdUser().then(
    (codUser)=>{
      this.codUser=codUser;
      console.log(codUser)
    }
  );
  this.userService.getLatLng().subscribe(
    (data)=>{
      this.latLng=data;
    }
  )
}

ionViewDidLoad() {
  console.log('ionViewDidLoad CartPage');
}

ngOnInit() {
  let items = this.cartService.getCart();
  let selected = {};
  console.log(items)
  for (let obj of items) {

    if (selected[obj.codproducto]) {
      selected[obj.codproducto].count++;
    } else {
      selected[obj.codproducto] = {...obj, count: 1};
      console.log("objid :" +
JSON.stringify(selected[obj.codproducto]))
    }
  }
  this.productos = Object.keys(selected).map(key =>
selected[key])

  this.total = this.productos.reduce((a, b) => a + (b.count
* b.precio), 0);
  console.log(this.productos)
}

saveOrder(){

```



```

        let payment = new PayPalPayment(this.total+'.00',
        'USD', 'Pago por el pedido', 'sale');

this.payPal.renderSinglePaymentUI(payment).then((response) =>
{
    this.toastService.presentToast("Pago realizado
correctamente");
    this.saveOrder();
    }, () => {
        this.toastService.presentToast("Error al hacer el
pago de paypal");
    })
    })
    })
    }
    addDireccion(){
        this.navCtrl.push(MapPage);
    }
}
}

```

Figura N° 33: cart.ts

Fuente: Elaboración Propia

- Contenido del archivo entrega.html

```

<ion-header>
  <ion-navbar color="dark">
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Entregar</ion-title>
  </ion-navbar>
</ion-header>

<ion-content>
  <ion-list >
    <ion-list-header>Entregas Pendientes</ion-list-header>
    <ion-item *ngFor="let item of pedidos" >
      <ion-icon name="appstore" item-start></ion-icon>
      <h2>Pedido #{{item.codpedido.codpedido}}</h2>
      <p>Cliente: {{item.codpedido.codusuario.usuario}}</p>
      <p
*ngIf="!item.codpedido.latitud">{{item.codpedido.codusuario.d
ireccion}}</p>
      <p>{{item.codpedido.fecha.date}}</p>
      <p >{{item.codpedido.latitud}}--
{{item.codpedido.longitud}}</p>

```

```

        <button slot="end" ion-button outline item-end margin
(click)="entregarPedido(item)" >Entregar</button>
    </ion-item>
</ion-list>
</ion-content>

```

Figura N° 34: entrega.html

Fuente: Elaboración Propia

- Contenido del archivo entrega.ts:

```

import { Component } from '@angular/core';
import { NavController, NavParams, ViewController } from
'ionic-angular';
import { EntregaPedidoService } from '../services/services-
rest/entregaPedido.service';
import { EntregaPedido } from
'../models/entregaPedido.model';
import { ToastService } from '../services/toast.service';
import { SearchPage } from '../search/search';
import { UserStorageService } from '../services/user-
storage.service';
@Component({
  selector: 'page-entrega',
  templateUrl: 'entrega.html'
})
export class EntregaPage {
  selectedItem: any;
  icons: string[];
  items: Array<{title: string, note: string, icon: string}>;
  pedidos:EntregaPedido[];
  constructor(public navCtrl: NavController, public
navParams: NavParams,public
entregaPedidoService:EntregaPedidoService, public
toastService:ToastService,public
userStorageService:UserStorageService) {
    this.userStorageService.getIdUser().then(
      (codUsuario)=>{

this.entregaPedidoService.findByCodUsuario(codUsuario).subscr
ibe(
      (data)=>{
        this.pedidos=data['entregaPedido'];
        console.log(this.pedidos);
      }
    )
  }
}

```

```

    )
  }

  entregarPedido(){
    this.toastService.presentToast("Pedido entregado
correctamente");
  }
}

```

Figura N° 35: entrega.ts

Fuente: Elaboración Propia

- Contenido del archivo home.html

```

<ion-header>
  <ion-navbar color="dark">
    <button ion-button menuToggle start>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title text-center>
      Bienvenido
    </ion-title>
    <ion-buttons end>
      <cartcount></cartcount>
    </ion-buttons>
  </ion-navbar>
</ion-header>

<ion-content>

  <div *ngFor="let cat of items" class="category-block">

    <ion-row no-padding class="category-banner">
      <ion-col text-left button tappable
(click)="cat.expanded = !cat.expanded" align-self-center>
        {{ cat.category }}
      </ion-col>
    </ion-row>

    <ion-slides [options]="sliderConfig">
      <ion-slide *ngFor="let product of cat.empresas">
        <div *ngIf="cat.expanded">

          <ion-card>
            <ion-card-header>
              <ion-card-title>

```

```

        {{ product.nombre }}
    </ion-card-title>
    <ion-card-content>

        
        <!--<ion-button color="primary" expand="full"
(click)="addToCart(product)"></ion-button-->
        <button ion-button
(click)="verProductos(product)" >ver</button>

    </ion-card-content>
    </ion-card-header>
    </ion-card>
</div>
</ion-slide>
</ion-slides>

</div>
</ion-content>

```

Figura N° 36: home.html

Fuente: Elaboración Propia

- Contenido del archivo home.ts:

```

import { Component, OnInit } from '@angular/core';
import { NavController } from 'ionic-angular';
import { CartService } from '../services/cart.service';
import { ProductoPage } from '../pages/producto/producto';
import { Empresa } from '../models/empresa.model';
@Component({
  selector: 'page-home',
  templateUrl: 'home.html',
})
export class HomePage implements OnInit {
  cart = [];
  items = [];

  sliderConfig = {
    slidesPerView: 1.6,
    spaceBetween: 10,
    centeredSlides: true
  };

  constructor(public navCtrl: NavController, private
cartService: CartService) {

```

```

    }
    ngOnInit() {
      this.items = this.cartService.getProducts();
      this.cart = this.cartService.getCart();
    }

    addToCart(product) {
      this.cartService.addProduct(product);
    }
    verProductos(empresa: Empresa) {
      this.navCtrl.push(ProductoPage, {
        codempresa: empresa.codempresa, nombre: empresa.nombre});
      console.log()
    }
  }
}

```

Figura N° 37: home.ts

Fuente: Elaboración Propia

- Contenido del archivo map.html

```

<ion-header>
  <ion-navbar color="dark">
    <ion-title text-center>
      Buscar dirección
    </ion-title>
    <ion-buttons end>

    </ion-buttons>
  </ion-navbar>
</ion-header>
<ion-content padding>
  <ion-input type="text" placeholder="dirección"
name="direccion" [(ngModel)]="direccion" required></ion-
input>
  <button ion-button color="primary"
(click)="geocoder()">buscar</button>
  <button ion-button color="primary"
(click)="back()">Agregar dirección</button>
  <div id="map_canvas"></div>
</ion-content>

```

Figura N° 38: map.html

Fuente: Elaboración Propia

- Contenido del archivo map.ts:

```

import {MapService} from '../services/map.service';
import { Component } from "@angular/core/";
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import {UserStorageService} from '../services/user-
storage.service';
@Component({
  selector: 'page-map',
  templateUrl: 'map.html'
})
export class MapPage {
  direccion='Av. Peru 700 andahuaylas';
  constructor(public mapService:MapService,public
navController:NavController,public
userStorageService:UserStorageService) { }

  ionViewDidLoad() {
    this.mapService.loadMap('map_canvas');
  }
  geocoder(){
    this.mapService.geocoder(this.direccion).then(
      (data)=>{
        console.log("valor de la promesa");
        console.log(data);
        if(data.length){
          const
latLng={lat:data[0].position.lat,lng:data[0].position.lng,dir
eccion:this.direccion}
          this.userStorageService.setLatLng(latLng);
        }
      }
    )
  }
  back(){
    this.navController.pop();
  }
}

```

Figura N° 39: map.ts

Fuente: Elaboración Propia

- Contenido del archivo producto.html

```

<ion-header>
  <ion-navbar color="dark">
    <ion-title text-center>
      {{nombreEmpresa}}

```

```

    </ion-title>
    <ion-buttons end>
      <cartcount></cartcount>
    </ion-buttons>
  </ion-navbar>
</ion-header>

<ion-content>
  <!-- <section padding margin-bottom>
    <ion-icon name="person"></ion-icon>
    <h1 style="font-size:1.4em" no-margin>Welcome, <br>
      Mr. John Doe</h1>
  </section-->
  <ion-grid>
    <ion-row wrap>
      <ion-col width=50 *ngFor="let item of productos">
        <ion-card >
          
          <ion-card-content>
            <ion-card-title
class="cardbody">{{item.titulo}}</ion-card-title>
            <p ion-text
color="light">{{item.descripcion}} </p>
            <p ion-text color="light">Precio:
{{item.precio}} </p>
            <button ion-button margin
(click)="addToCart(item)" >Agregar</button>
          </ion-card-content>
        </ion-card>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>

```

Figura N° 40: producto.html

Fuente: Elaboración Propia

- Contenido del archivo producto.ts:

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { ProductoService } from '../services/services-
rest/producto.service';
import { Producto } from '../models/producto.model';
import { CartService } from '../services/cart.service';

```

```

/**
 * Generated class for the ProductoPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation
for more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-producto',
  templateUrl: 'producto.html',
})
export class ProductoPage {
  private productos:Producto[]=[];
  private nombreEmpresa:String="";
  constructor(public navCtrl: NavController, public
navParams: NavParams, public
productoService:ProductoService,private cartService:
CartService) {
    const codEmpresa= this.navParams.get("codempresa");
    this.nombreEmpresa= this.navParams.get("nombre");
    console.log(codEmpresa)

this.productoService.findByCodEmpresa(codEmpresa).subscribe(
  (data)=> {
    this.productos=data['productos'];
    console.log(this.productos)
  }
)
}

ionViewDidLoad() {
  console.log('ionViewDidLoad ProductoPage');
}
addToCart(product) {
  this.cartService.addProduct(product);
}
}

```

Figura N° 41: producto.ts

Fuente: Elaboración Propia

- Contenido del archivo register.html

```

<ion-content class="register-content" padding>
  <ion-row class="logo-row">
    <ion-col></ion-col>
    <ion-col class="logo">
    </ion-col>
    <ion-col></ion-col>
  </ion-row>
  <div class="register-box">
    <form #registerForm="ngForm">
      <ion-row>
        <ion-col>
          <ion-list inset>
            <ion-item>
              <ion-icon name="contact" item-start></ion-icon>
              <ion-input type="text" placeholder="Usuario"
name="usuario" [(ngModel)]="credentials.usuario"
required></ion-input>
            </ion-item>
            <ion-item>
              <ion-icon name="lock" item-start></ion-icon>
              <ion-input type="password"
placeholder="Contraseña" name="contrasena"
[(ngModel)]="credentials.contrasena" required></ion-input>
            </ion-item>
            <ion-item>
              <ion-icon name="home" item-start></ion-icon>
              <ion-input type="text" placeholder="Dirección"
name="direccion" [(ngModel)]="credentials.direccion"
required></ion-input>
            </ion-item>
            <ion-item>
              <ion-icon name="call" item-start></ion-icon>
              <ion-input type="text" placeholder="Telefono"
name="telefono" [(ngModel)]="credentials.telefono"
required></ion-input>
            </ion-item>
          </ion-list>
        </ion-col>
      </ion-row>
      <ion-row>
        <ion-col class="signup-col">
          <button ion-button class="submit-btn" full
(click)="register()"
[disabled]="!registerForm.form.valid">Crear cuenta</button>

```

```

        <button ion-button class="register-btn" block clear
(click)="login()">Iniciar sesión</button>
    </ion-col>
</ion-row>
</form>
</div>
</ion-content>

```

Figura N° 42: register.html

Fuente: Elaboración Propia

- Contenido del archivo register.ts:

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { User } from '../models/user.model';
import { LoginPage } from '../pages/login/login';
import { UserService } from '../services/services-
rest/user.service';
import { ToastService } from '../services/toast.service';
import { Strings } from "../utils/strings";
@IonicPage()
@Component({
  selector: 'page-register',
  templateUrl: 'register.html',
})
export class RegisterPage {
  credentials: User = { usuario: '', contraseña: '',
direccion: '', telefono: 0, rol: '' };
  constructor(public navCtrl: NavController, public
navParams: NavParams, public userService: UserService, public
toastService: ToastService) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad RegisterPage');
  }

  register(){
    this.userService.register(this.credentials).subscribe(
      (data)=>{

this.toastService.presentToast(Strings.OPERACION_EXITOSA);
      setTimeout(() => this.navCtrl.setRoot(LoginPage),
2000);
    },

```

```

        (error)=> {
            console.log(error);
        }
    );
}
login(){
    this.navCtrl.push(LoginPage);
}
}

```

Figura N° 43: register.ts

Fuente: Elaboración Propia

- Contenido del archivo search.html

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-searchbar placeholder="Buscar"
      [(ngModel)]="searchTerm" [formControl]="searchControl"></ion-
searchbar>
    <ion-buttons end>
      <cartcount></cartcount>
    </ion-buttons>
  </ion-navbar>
</ion-header>
<ion-content padding>
  <ion-list *ngIf="toggle" >
    <ion-list-header>BUSQUEDA</ion-list-header>
    <ion-item *ngFor="let item of autocompleteItems" >
      <ion-avatar item-start>
        
      </ion-avatar>
      <h2>{{item.titulo}}</h2>
      <p>{{item.descripcion}}</p>
      <p>precio s/.{{item.precio}}</p>
      <button ion-button item-end margin
        (click)="addToCart(item)" >Agregar</button>
    </ion-item>
  </ion-list>
</ion-content>

```

Figura N° 44: search.html

Fuente: Elaboración Propia

- Contenido del archivo search.ts:

```
import { Component,OnInit } from '@angular/core';
import {NavController, NavParams } from 'ionic-angular';
import { FormControl } from '@angular/forms';
import {ProductoService } from '../services/services-rest/producto.service';
import { CartService } from '../services/cart.service';
import 'rxjs/add/operator/debounceTime';
@Component({
  selector: 'page-search',
  templateUrl: 'search.html',
})

export class SearchPage implements OnInit{
  autocompleteItems=[];
  searchTerm: string = '';
  searchControl: FormControl;
  toggle: boolean = false;
  temp: any;
  constructor(public navCtrl: NavController, public
navParams: NavParams, private
productoService:ProductoService,private
cartService:CartService) {
    this.searchControl = new FormControl();

this.searchControl.valueChanges.debounceTime(700).subscribe(s
earch => {
  if(this.temp){
    this.autocompleteItems=this.temp;
    this.findAll();
  }
  this.filterItems();
});
}
ngOnInit() {
  this.findAll();
}
findAll(){
  this.productoService.findAll().subscribe(
(data)=>{
    if(this.autocompleteItems.length==0){
      this.autocompleteItems = data['productos'];
    }
    this.temp=data['productos'];
    console.log(this.temp);
  }
}
```

```

    );
  }
  filterItems() {
    console.log("ejecutado filter")
    this.toggle = true;
    if (!this.searchTerm) {
      this.toggle = false;
    }
    this.autocompleteItems =
this.autocompleteItems.filter((item) => {
  return
((item.titulo).toLowerCase().indexOf(this.searchTerm.toLowerCase()) > -1);
})
  }
  ionViewDidLoad() {
    console.log('ionViewDidLoad SearchPage');
  }
  addToCart(product) {
    this.cartService.addProduct(product);
  }
}

```

Figura N° 45: search.ts

Fuente: Elaboración Propia

3.4.5. Fase de Pruebas

En esta última fase de la metodología Mobile-D, se realiza un conjunto de pruebas de la aplicación móvil, se identifica y se soluciona algunos errores, esta fase busca entregar una versión completamente estable y funcional de la aplicación móvil. Una vez que se solucionaron todos los errores identificados, recién se puede obtener el entregable final estable.

Al terminar el desarrollo de los módulos de la aplicación móvil se pasó a esta fase de pruebas, donde se iteró, para obtener un módulo funcional y al final de todo se llegó a la versión estable.

3.5. Administración y Gestión del Proyecto Basado en PMBOOK

Para la elaboración de este proyecto se tomará como referencia el PMBOOK (Project Management Body of Knowledge) quinta

edición, que es una guía que contiene el cuerpo de conocimiento para desarrollar de manera profesional la gerencia de proyectos.

3.5.1. Plan de Gestión de la Integración

Tabla N° 25: Plan de Gestión de la Integración

PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PATROCINADO POR	Juan Carlos Palomino Miranda
REVISIÓN (Correlativo)	DESCRIPCIÓN (REALIZADA POR)
01	Elaboración de Acta de Constitución (Juan Carlos Palomino Miranda)
BREVE DESCRIPCIÓN DE LA APLICACIÓN MÓVIL DEL PROYECTO	
Este proyecto tiene como objetivo desarrollar una aplicación móvil multiplataforma, para la gestión de pedidos de comida vía delivery en los distritos de Andahuaylas, Talavera y San Jerónimo, para que los clientes de los diferentes restaurantes, pollerías, pizzerías, recreos, etc. Puedan degustar de los diferentes platillos que ofrecen estos locales en la comodidad de su hogar, sin la necesidad de acudir al establecimiento.	
ALINEAMIENTO DEL PROYECTO	
OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN	PROPÓSITO DEL PROYECTO
Desarrollar aplicaciones multiplataforma con	Impulsar la venta de comida vía delivery a través de una aplicación móvil, en los

bajo costo de producción.	distritos de Andahuaylas, Talavera y San Jerónimo.
OBJETIVOS DEL PROYECTO	
<ul style="list-style-type: none"> - Desarrollar un módulo que permita mostrar la carta ofrecida por cada establecimiento. - Desarrollar una aplicación móvil multiplataforma que permita gestionar comidas vía delivery utilizando un teléfono inteligente. - Se desarrollará una función agregar producto para posteriormente el usuario pueda listar pedidos donde se de la información del precio total de los productos seleccionados. 	
FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO	
<ul style="list-style-type: none"> - Adquisición de hardware y software dentro de las fechas establecidas y con las características adecuadas. - Diseño y desarrollo de la aplicación móvil con mejoras en la gestión de pedidos de comida vía delivery. 	
REQUERIMIENTOS DE ALTO NIVEL	
<ul style="list-style-type: none"> - Exploración de las herramientas para el proyecto. - Diseñar e integrar los módulos y servicios de la aplicación. - La aplicación móvil se desarrollará siguiendo las fases de la metodología Mobile – D. 	
EXTENSION Y ALCANCE DEL PROYECTO	
FASES DEL PROYECTO	PRINCIPALES ENTREGABLES
1. Gestión del Proyecto	Acta de Constitución del Proyecto.
2. Diseño	Definición de las diferentes interfaces de la aplicación móvil.

3.Requerimiento y Adquisiciones	Adquisición de Laptop y Smatphones. Proceso de Instalación de los diferentes softwares de desarrollo (Ionic 3, Visual Studio Code, Node JS y otros).
4. Implementación	Implementar los diferentes módulos de la aplicación móvil.
5. Arranque del Sistema	Pruebas de la Aplicación Móvil.
6. Término de Obra	Aplicación Móvil Multiplataforma.
INTERESADOS CLAVE	
<ul style="list-style-type: none"> - Clientes de los diferentes establecimientos de comida en los distritos de Andahuaylas, Talavera y San Jerónimo. - Dueños de los diferentes establecimientos de comida de los distritos de Andahuaylas, Talavera y San Jerónimo que deseen optimizar la gestión de pedidos de comida vía delivery. - Diferentes programadores y desarrolladores de aplicaciones multiplataforma. 	
RIESGOS	
<ul style="list-style-type: none"> - Atraso en la adquisición de hardware y software necesarios para el desarrollo del proyecto. - Cambio de políticas y licencias de los diferentes entornos y frameworks. - Recursos económicos insuficientes. 	
HITOS PRINCIPALES DEL PROYECTO	
<ul style="list-style-type: none"> - Buena PRO y aprobación del acta de constitución del Proyecto - Lograr con la implementación de la aplicación móvil multiplataforma y cumplir con todos los objetivos trazados. 	

- Realización de pruebas y correcciones en la aplicación móvil implementada.
--

PRESUPUESTO DEL PROYECTO

El costo del proyecto será asumido por el Jefe del Proyecto: Juan Carlos Palomino Miranda

Fuente: Elaboración Propia

3.5.2. Plan de Gestión del Alcance

Tabla N° 26: Plan de Gestión del Alcance

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO	Juan Carlos Palomino Miranda
Describir cómo será administrado el Alcance del Proyecto	
<p>En primer lugar, se estableció, para qué tipo de usuarios estará dirigido el proyecto.</p> <p>En segundo lugar, se determinó el alcance del proyecto, es que la aplicación móvil desarrollada será utilizada por los diferentes clientes de los diferentes establecimientos de comida de los distritos de Andahuaylas, Talavera y San Jerónimo.</p>	
Evaluar la estabilidad del alcance del proyecto (Cómo manejar los cambios, la frecuencia e impacto de los mismos):	
<p>Para poder realizar cambios en el proyecto, estos debes ser solicitados, evaluados y aprobados por el Jefe del Proyecto.</p> <p>Cuando se realice una solicitud de cambio, el Jefe del Proyecto deberá analizar el impacto que generará los cambios solicitados en el Proyecto.</p>	

Se tomará un día a la semana para poder realizar reuniones en donde se debatirán y analizarán las solicitudes de cambio que surgirán durante el desarrollo del Proyecto.

¿Cómo los cambios al alcance, serán identificados y clasificados?:

El Jefe del Proyecto es el encargado de revisar, analizar las diferentes solicitudes de cambio del alcance del Proyecto, para posteriormente identificar y clasificar los cambios solicitados, de la misma manera podrá solicitar información adicional acerca de los cambios solicitados.

Describir cómo los cambios del alcance serán integrados al proyecto:

El Jefe del Proyecto deberá debe asegurar que el cambio solicitado no afecte la línea base del Proyecto, luego de ello se deberán actualizar las líneas base del Proyecto.

Comentarios Adicionales:

Si algún cambio solicitado para el alcance del Proyecto afecta de manera perjudicial al Proyecto, este no será tomado en cuenta por el Jefe del Proyecto.

Fuente: Elaboración Propia

3.5.3. Plan de Gestión del Tiempo

Tabla N° 27: Plan de Gestión del Tiempo

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.	
PREPARADO POR	Juan Carlos Palomino Miranda	
Persona(s) autorizada(s) a solicitar cambio en cronograma:		
NOMBRE	CARGO	UBICACIÓN

Juan Carlos Palomino Miranda	Jefe del Proyecto	Andahuaylas
Persona(s) que aprueba(n) el requerimiento de cambio de cronograma:		
NOMBRE	CARGO	UBICACIÓN
Juan Carlos Palomino Miranda	Jefe del Proyecto	Andahuaylas
Razones aceptables para cambios en cronograma del Proyecto:		
<ul style="list-style-type: none"> • Fallo de herramientas de hardware (Laptop, Smartphone, otros) • Accidentes de trabajo • Errores en el establecimiento de secuencia de actividades. • Fallas y/o averías en la conexión a internet. 		
Describir cómo calcular y reportar el impacto en el proyecto por el cambio en cronograma:		
<ul style="list-style-type: none"> • Se realiza un informe al responsable del mismo con la fecha en el que ocurrió el problema. • El problema será descrito de manera detallada, de igual manera se deberá indicar el grado de urgencia. • Impacto del mismo sobre el proyecto (Tiempo, Alcance, Calidad y Costo). • Se deberá describir las diferentes alternativas de solución con detalles de los impactos en las diferentes áreas(Tiempo, Alcance, Calidad y Costo). 		
Describir cómo los cambios al cronograma serán administrados:		
<p>El alcance del Proyecto se llevará acabo de la siguiente manera:</p> <p>Designación de Responsabilidades:</p>		

- Planificación (Jefe del Proyecto).
- Seguimiento y Control (Jefe del Proyecto).
- Ejecución (Jefe del Proyecto).

Fuente: Elaboración Propia

3.5.4. Plan de Gestión de Costos

Tabla N° 28: Plan de Gestión de Costos

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.	
PREPARADO, REVISADO Y APROBADO POR	Juan Carlos Palomino Miranda	
Persona(s) autorizada(s) a solicitar cambios en el costo:		
NOMBRE	CARGO	UBICACIÓN
Juan Carlos Palomino Miranda	Jefe del Proyecto	Andahuaylas
Persona(s) que Aprueba(n) requerimientos de cambios en costo contractual:		
NOMBRE	CARGO	UBICACIÓN
Juan Carlos Palomino Miranda	Jefe del Proyecto	Andahuaylas
Persona(s) que aprueba(n) el requerimiento de cambio de costo interno ofrecido:		
<ul style="list-style-type: none"> • Jefe del Proyecto 		
Razones aceptables para cambios en el Costo del Proyecto:		

- Mejoras admisibles en el Alcance del Proyecto.
- Aumento de plazos en el Alcance del Proyecto

Describir cómo calcular e informar el impacto en el proyecto por el cambio en el costo:

- Usuarios que soliciten cambios en el Proyecto.
- Determinar las características de la situación que requiera la solicitud de cambio.
- Determinar las diferentes alternativas de solución, haciendo mención las diferentes áreas como el tiempo, alcance, calidad y costo.
- Selección de alternativas de solución.

Fuente: Elaboración Propia

3.5.5. Plan de Gestión de Calidad

Tabla N° 29: Plan de Gestión de Calidad

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO, REVISADO, APROBADO POR	Juan Carlos Palomino Miranda
GESTIÓN DE CALIDAD DEL PROYECTO	
PLANIFICAR LA CALIDAD	

En el plan de gestión de calidad se describe que el equipo de dirección será el encargado de implementar las políticas de calidad de la organización que ejecuta. Este es un plan subsidiario o un componente del plan para la dirección del proyecto.

Dicho plan de calidad proporciona las entradas al plan general para la dirección del proyecto e integra el control de calidad, el aseguramiento de la calidad y los métodos que mejoran los procesos del proyecto.

Para esto se aplicará, la calidad de software (SQA, Software Quality Assurance), el cual no es más que un conjunto de actividades previamente planificadas y sistemáticas necesarias para contribuir la confianza en que el software satisfaga los requerimientos de calidad, este proceso será aplicado durante todo el desarrollo del software.

REALIZAR ASEGURAMIENTO DE CALIDAD

La ejecución del Aseguramiento de Calidad del Proyecto, durante el desarrollo del Proyecto, será asumido por el Jefe del Proyecto, el cual se encarga de supervisar el planeamiento de los procesos del proyecto, plantea acciones correctivas o preventivas según se requiera. Estos procesos se deberán informar cada semana a todo el Equipo del Proyecto

REALIZAR CONTROL DE CALIDAD

Realizar la ejecución del Control de Calidad será responsabilidad del Jefe del Proyecto, deberá revisar los diferentes entregables del proyecto según se vayan presentando luego se deberá emitir conformidades y/o observaciones de los mismos en las reuniones programadas durante la semana.

Tendremos dos procedimientos para llevar el control de calidad:

- Revisión de Forma
- Revisión de Contenidos

CONTROL DE CALIDAD DEL CONTENIDO: Se revisa la calidad de los entregables del proyecto con el equipo de expertos.

CONTROL DE CALIDAD DE REDACCIÓN Y FORMATO: Se revisa la redacción de los documentos y las observaciones que se puedan realizar.

Fuente: Elaboración Propia

3.5.6. Plan de Gestión de Recursos Humanos

Tabla N° 30: Plan de Gestión de Recursos Humanos

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO, REVISADO, APROBADO POR	Juan Carlos Palomino Miranda
ROL EN EL PROYECTO	JEFE DEL PROYECTO
FUNCION / RESPONSABILIDAD PRINCIPAL	
<ul style="list-style-type: none"> • El Jefe del Proyecto será el principal responsable de la ejecución y entrega final del proyecto. • Responsable de la designación del personal que está inmerso en el equipo del Proyecto y de la Organización. • Control de los diferentes costos y gastos del Proyecto. 	
COMPETENCIAS REQUERIDAS / RESPONSABILIDADES (Conocimientos, habilidades, actitudes)	
DISPONIBILIDAD	

Puesto que se requiere	Interno / Externo	Fecha Ingreso	Fecha Salida	Tiempo Requerido
Jefe del Proyecto	Interno	01/03/2019	30/06/2019	4 meses
INCORPORACIÓN AL PROYECTO				
EVENTO DISPARADOR		SINCRONIZACIÓN PREVISTA		
RENUNCIA		Al instante con la designación de reemplazo por el Jefe del Proyecto.		
ENFERMEDAD		Al instante con la designación de reemplazo por el Jefe del Proyecto.		

Fuente: Elaboración Propia

3.5.7. Plan de Gestión de Comunicaciones

Tabla N° 31: Plan de Gestión de Comunicaciones

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO, REVISADO, APROBADO POR	Juan Carlos Palomino Miranda
GESTIÓN DE COMUNICACIONES DEL PROYECTO (Descripción de cómo se van a aplicar los procesos de gestión de comunicaciones del proyecto, herramientas a emplear, responsables, áreas de aplicación, etc.)	
Será responsabilidad el Jefe del Proyecto integrar la Base de Datos, Plugins, Módulos, etc.	

PLANIFICACIÓN DE LA COMUNICACIÓN

Determinar la información requerida, necesidades y comunicación de los interesados: que información, quien la solicita, para cuándo la requiere, cómo y quién será el encargado de suministrar dicha información.

El jefe del Proyecto deberá prever la cantidad y calidad de canales requeridos desde el principio con la finalidad de que la información fluya y llegue a todo el personal involucrado.

Requisitos para la comunicación incluyen:

- Relaciones de compromiso dentro de la Organización
- Organigramas

Fuente: Elaboración Propia

3.5.8. Plan de Gestión de Riesgos

Tabla N° 32: Plan de Gestión de Riesgos

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO, REVISADO, APROBADO POR	Juan Carlos Palomino Miranda
Descripción de la metodología de gestión del riesgo a ser usada:	
ALCANCES:	<ul style="list-style-type: none">• La priorización, seguimiento e identificación de los principales riesgos críticos estará a cargo del Jefe el Proyecto.
HERRAMIENTAS:	

<ul style="list-style-type: none"> • Se utilizará la lluvia de ideas.
<p>FUENTES DE DATOS:</p> <ul style="list-style-type: none"> • Los integrantes del proyecto serán los encargados de identificar los riesgos según juicio y experiencia.
<p>ROLES Y RESPONSABILIDADES:</p> <p>JEFE DE PROYECTO: Tendrá como responsabilidad identificar, priorizar y hacer el seguimiento correspondiente de los riesgos que puedan afectar en el desarrollo del proyecto, también será quien proponga acciones para resolver los riesgos debidamente identificados.</p> <p>ANALISTA DE SISTEMAS: Será el encargado de la determinación de la funcionalidad de la aplicación móvil desarrollada.</p> <p>PROGRAMADOR: Será el encargado de realizar toda la programación e integración en lo que dure el desarrollo de la aplicación móvil.</p>

Fuente: Elaboración Propia

3.5.9. Plan de Gestión de Adquisiciones

Tabla N° 33: Plan de Gestión de Adquisiciones

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO, REVISADO, APROBADO POR	Juan Carlos Palomino Miranda
<p>RECURSOS PARA LA ADQUISICIÓN:</p> <p>En este proyecto los que serán en cargados de realizar la contratación y labores de compra son:</p> <ul style="list-style-type: none"> • El Jefe de Proyecto en la parte de adquisición de herramientas de desarrollo y equipos físicos. 	

- El Jefe del Proyecto en el lado de contrataciones (programador, analista).

En el presente proyecto el Jefe del Proyecto, Analista de Sistemas, Programador, Diseñador de interfaces de la aplicación móvil estarán a cargo de una sola persona.

PRODUCTOS Y SERVICIOS CONTRATADOS:

- Servicio del proveedor para la compra de los equipos físicos.
- Servicio de desarrollador para la aplicación móvil.

Procedimientos para la gestión de adquisiciones

PROCEDIMIENTO EN EL PROYECTO PARA SOLICITUD DE COMPRA

En este documento se describirá el producto que se requiere y la cantidad en unidades.

Luego se deberá evaluar, para posteriormente el comprador de la empresa envía los requerimientos a los proveedores y estos devolverán una plantilla para la toma de decisión económica y técnica que se ajuste de mejor manera a los requerimientos del proyecto.

PROCEDIMIENTO EN EL ÁREA LOGÍSTICA PARA REALIZAR ORDEN DE COMPRA

Se elabora la Orden de Compra: En este documento se debe anotar el nombre completo del proveedor, fecha, nombre de la persona de contacto (si la hay), número telefónico, detalle del producto a pedir, cantidad, precio unitario y monto total.

Fuente: Elaboración Propia

3.5.10. Plan de Gestión de Los Interesados

Tabla N° 34: Plan de Gestión de los Interesados

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Gestión de Pedidos de Comida Vía Delivery en los Distritos de Andahuaylas, Talavera y San Jerónimo.
PREPARADO POR	Juan Carlos Palomino Miranda
REVISADO PO	Juan Carlos Palomino Miranda
APROBADO POR	Juan Carlos Palomino Miranda

NOMBRES Y APELLIDOS	CARGO	INFORMACIÓN DEL CONTACTO	REQUERIMIENTOS SOBRE EL PRODUCTO	INFLUENCIA SOBRE	TIPO DE INTERÉS
Juan Carlos Palomino Miranda	Jefe del Proyecto	Admjc02@gmail.com	Responsable de brindar la viabilidad, disponibilidad de todos los recursos que sean necesarios para ejecución del proyecto.	Estará presente en todas la fases del desarrollo del proyecto.	Control del Proceso
Juan Carlos Palomino Miranda	Analista de Sistemas	Admjc02@gmail.com	Responsable del diseño de la infraestructura, interfaz visual de la aplicación móvil. Para ello deberá considerar los requisitos funcionales, el alcance y limitaciones que la aplicación móvil tiene.	Fases de Planificación, Ejecución y Cierre	Elaborar la infraestructura de la aplicación y dar solución a los problemas que se presentan.
Juan Carlos Palomino Miranda	Programador de Sistemas	Admjc02@gmail.com	Responsable de la implementación e integración de los módulos que son diseñados por el Analista de Sistemas. De la misma manera será el encargado de realizar los test de prueba de la aplicación móvil.	Fases de Planificación, Ejecución y Cierre	Desarrollar la aplicación móvil en el cronograma que se estableció.

Fuente: Elaboración Propia

4. CONCLUSIONES

Conclusión general:

La metodología utilizada en este proyecto fue Mobile-D la cual permitió agilizar y organizar el trabajo, gracias a esta metodología se logró desarrollar la aplicación móvil con éxito, ya que es muy práctico, elástico y adaptable en los procesos de desarrollo de aplicaciones móviles, esto permitió tener una aplicación móvil estable y escalable.

Conclusiones Específicas:

1. Con el uso del framework Ionic se logró desarrollar un módulo en la aplicación móvil, la cual permita mostrar la carta ofrecida por cada establecimiento de comida registrada en la aplicación móvil-
2. El desarrollo de aplicaciones multiplataforma va creciendo a pasos agigantados dentro de la industria de desarrollo software, es por ello que al utilizar los plugins de cordova disponibles para ser utilizados con Ionic en este proyecto, se puede implementar una aplicación móvil multiplataforma.
3. Se creó una clase llamada "car.service.ts", en la cual se implementó funciones de agregar producto, eliminar producto, listar pedido las cuales son necesarias para que el cliente concrete el pedido.

5. RECOMENDACIONES

1. Se recomienda en un futuro se logre implementar otros medios de pago medios de pago vía tarjeta de crédito (VISA) esto con la finalidad de darle comodidad al cliente al momento de pagar.
2. Cuando se pretenda desarrollar aplicaciones de este tipo o cualquier otro, el desarrollador debe leer la documentación del framework que va utilizar, esto con la finalidad de conocer con precisión qué es lo que hacen.
3. Cuando se implementa módulos y métodos el desarrollador debe ir probando posibles errores, bugs, etc. Para que se corrija de manera inmediata.

6. REFERENCIAS BIBLIOGRÁFICAS

- Amaya Balaguera, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado Actual. *Journal Technology*, 111-124.
- Apache Cordova. (15 de Marzo de 2018). *Apache Cordova*. Obtenido de Apache Cordova: <https://cordova.apache.org/docs/es/latest/guide/overview/>
- Atiencia Pogo, K. L. (2017). *Aplicación móvil multiplataforma que brinda información de las unidades de salud públicas y privadas de la ciudad de Loja*. Ecuador: Universidad Nacional de Loja.
- Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodríguez, P. (30 de Agosto de 2009). *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone*. Obtenido de <http://www.adamwesterski.com>: http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf
- Cajilima Alvarado, J. R. (2015). *Desarrollo de una Aplicación, para Dispositivos Móviles que Permita Administrar y Controlar Rutas de los Vendedores, Aplicada a la Empresa "ALMACENES JUAN ELJURI CÍA.LTDA."* División Perfumería. Cuenca - Ecuador: Universidad Politécnica SALESIANA.
- Cuello, J., & Vittone, J. (26 de Julio de 2013). *Googole Books*. Obtenido de https://books.google.com.pe/books?id=ATiqsjH1rvwC&printsec=frontcover&hl=es&source=gbg_summary_r&cad=0#v=onepage&q&f=false
- Daneri Beltrán, S. G. (2017). *Un Framework Ágil para realizar Pruebas Funcionales de Software orientado a Aplicaciones Móviles, Caso: Aplicación Móvil para Transferencias Bancarias desarrollada bajo Metodología Scrum*. Lima: UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.
- Fernández Luna , J. M. (Septiembre de 2004). *Decsai* . Obtenido de Decsai : http://leo.ugr.es/J2ME/INTRO/intro_5.htm
- Fernández Luna , J. M. (Septiembre de 2004). *Decsai* . Obtenido de Decsai : http://leo.ugr.es/J2ME/INTRO/intro_6.htm
- Fernández Luna , J. M. (Septiembre de 2006). *Decsai*. Obtenido de Decsai: http://leo.ugr.es/J2ME/INTRO/intro_4.htm
- Hott Vidal, K. D., & Toro Oyorzún, S. A. (2011). *Sistema para la Implementación Masiva de Delivery Online de Comida*. VALPARAÍSO-CHILE: Universidad Técnica Federico Santa María.

- Huamantingo Navarro, R. R. (2018). *Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real*. Andahuaylas - Perú: Universidad Nacional José María Arguedas.
- Ionic Framework. (2017). *Ionic*. Obtenido de Ionic: <https://ionicframework.com/>
- Letelier, P., & Penadés, C. (2005). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Universidad Politécnica de Valencia*, 1-13.
- Meneses Sánchez, J. D., & Laveriano Meca, E. C. (2016). *Prototipo de aplicación móvil utilizando la metodología Mobile-D para la verificación de la formalidad en el servicio de taxi metropolitano en la ciudad de Lima*. Lima -Perú: Universidad Nacional Mayor de San Marcos.
- Pérez Porto, J., & Gardey, A. (14 de Marzo de 2018). *Definicion De*. Obtenido de Definicion De: <https://definicion.de/delivery/>
- Rangel Villagrán, I. M. (2013). *Procesos en el Desarrollo de Aplicaciones para Dispositivos Móviles*. México: Universidad Nacional Autónoma de México.
- Ribas Arbós, E. (29 de Mayo de 2018). *Business & Tech*. Obtenido de Business & Tech: <https://www.iebschool.com/blog/que-es-api-rest-integrar-negocio-business-tech/>
- Rojas Alegre, D. A. (2018). *Aplicación Móvil para Optimizar la Búsqueda de Productos y Servicios Utilizando Geolocalización en la Ciudad de Huaraz en el 2018*. Huaraz - Perú: Universidad Nacional Santiago Antúnez de Mayolo.
- Rouse, M. (21 de Enero de 2015). *TechTarget*. Obtenido de TechTarget: <https://searchdatacenter.techtarget.com/es/definicion/MySQL>
- Santiago Campi3n, R., Tralbaldo, S., Kamijo, M., & Fern3ndez, A. (2015). *Mobile Learning Nuevas Realidades en el Aula*. Espa1a: Editorial Oceano.
- Techopedia. (24 de Junio de 2017). *Techopedia*. Obtenido de Techopedia: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>

7. ANEXOS

MANUAL DE USUARIO

A continuación, se da una demostración del uso y funcionamiento de la aplicación WASIMAN.

- Las interfaces disponibles después de abrir la aplicación son las siguientes:
 - **INICIAR SESIÓN:** Permite al usuario autenticarse para que posteriormente pueda visualizar los diferentes establecimientos de comida.
 - **CREAR NUEVA CUENTA:** Lugar donde se registran datos del nuevo usuario y se crea la cuenta de acceso.
 - **OLVIDASTE LA CONTRASEÑA:** Opción que permite al usuario reestablecer la contraseña.

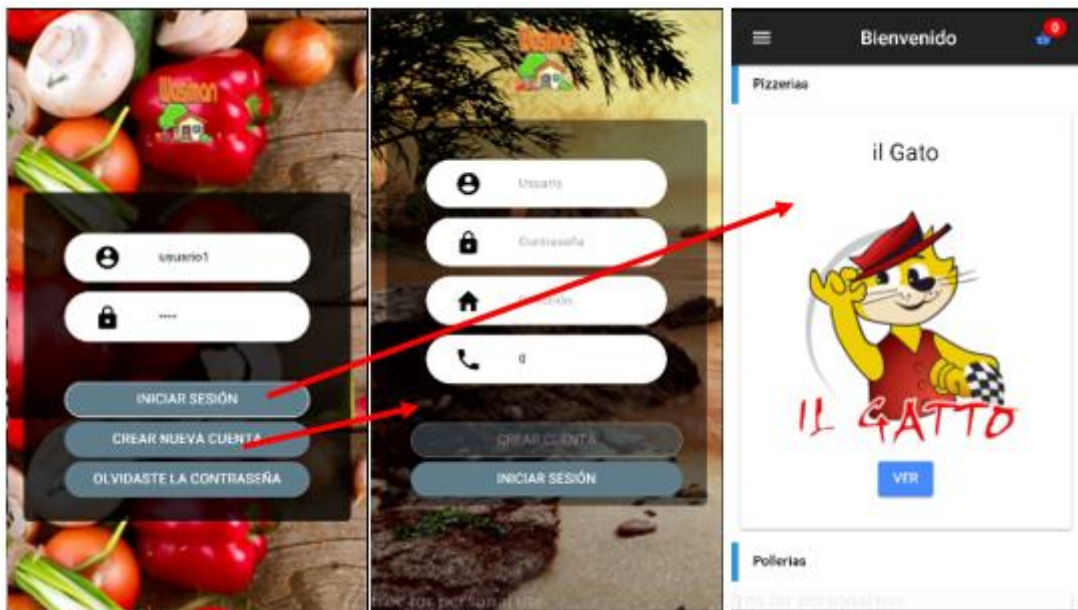


Figura N° 46: Acceso a la aplicación

Fuente: Elaboración Propia

- Cuando el usuario inicia sesión de manera correcta en la aplicación, se accede al menú principal en donde podrá observar los diferentes establecimientos de comida por categorías.

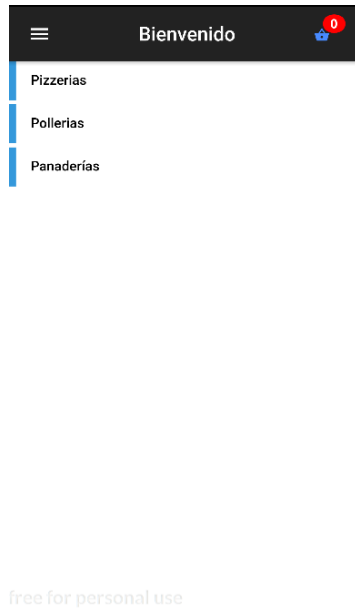


Figura N° 47: Menú Principal de la aplicación

Fuente: Elaboración Propia

- Una vez en el menú principal el usuario podrá escoger la categoría que desee. Por ejemplo, Pizzerías: en donde se le mostrará las diferentes pizzerías.

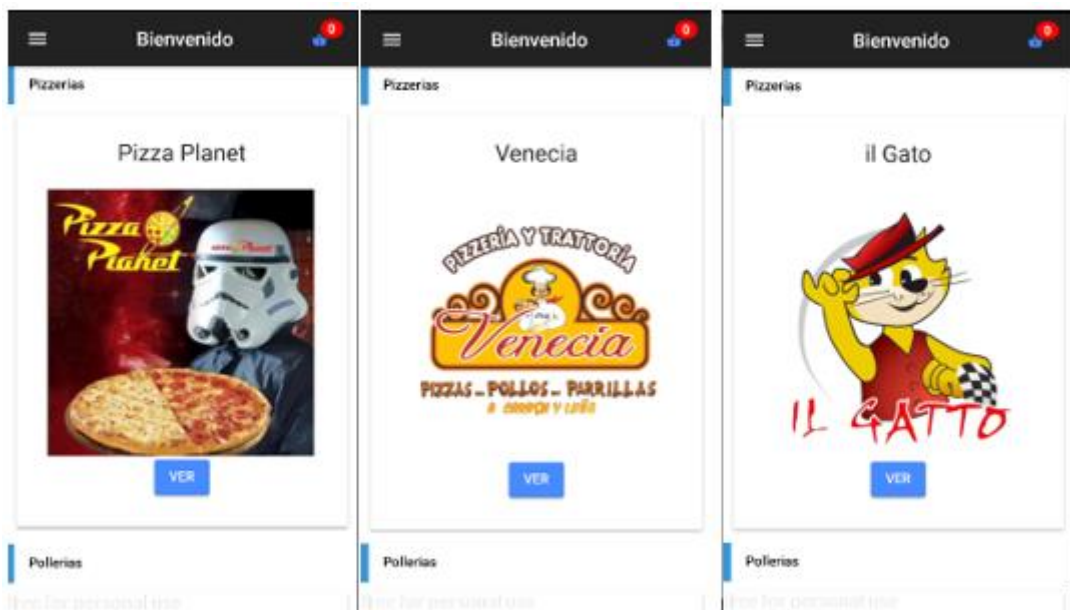


Figura N° 48: Categoría Pizzerías

Fuente: Elaboración Propia

- Una vez el usuario haya seleccionado la categoría, tendrá la opción de ver. Una vez seleccionado la opción ver se le mostrará los productos que ofrece el establecimiento seleccionado.



Figura N° 49: Productos Establecimiento
Fuente: Elaboración Propia

- Una vez dentro del establecimiento la aplicación mostrará los productos disponibles al usuario y este podrá agregar al carrito de compras los productos que desee.

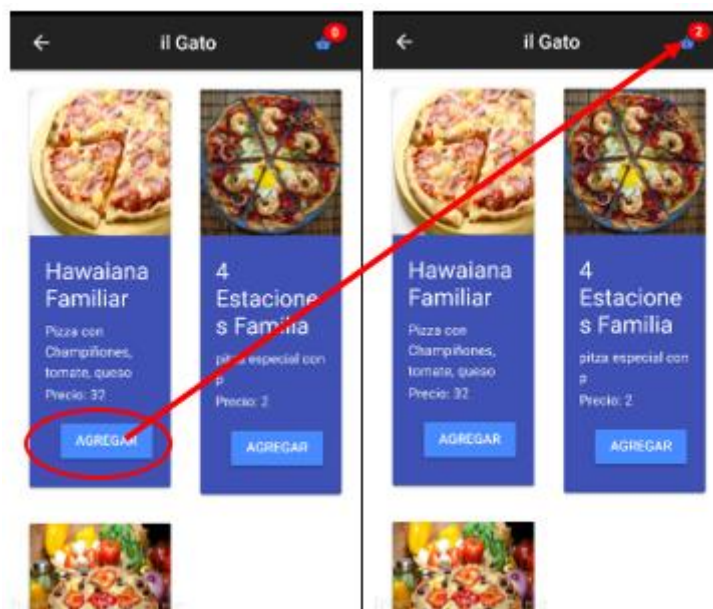


Figura N° 50: Agregar Productos al Carrito de Compras
Fuente: Elaboración Propia

- Cuando el usuario haya terminado de seleccionar todos los productos que desee de los diferentes establecimientos y esté listo para generar

el pedido, este deberá seleccionar el icono de carrito de compras que se encuentra en la parte superior derecha de la aplicación.

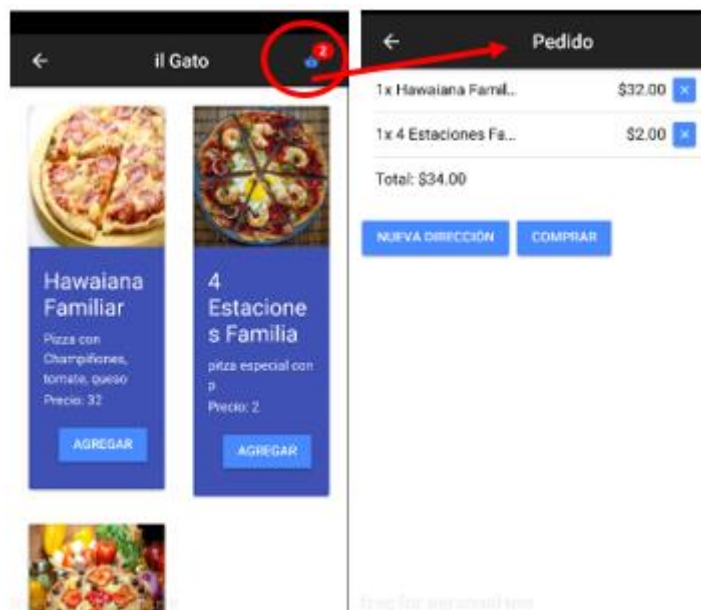


Figura N° 51: Carrito de Compras

Fuente: Elaboración Propia

- Una vez seleccionado la opción de carrito de compras el usuario estará listo para generar el pedido y realizar el pago correspondiente.
 - **NUEVA DIRECCIÓN:** Permite al usuario cambiar de dirección de entrega del pedido.
 - **COMPRAR:** Esta opción permitirá hacer el pago del pedido vía paypal.

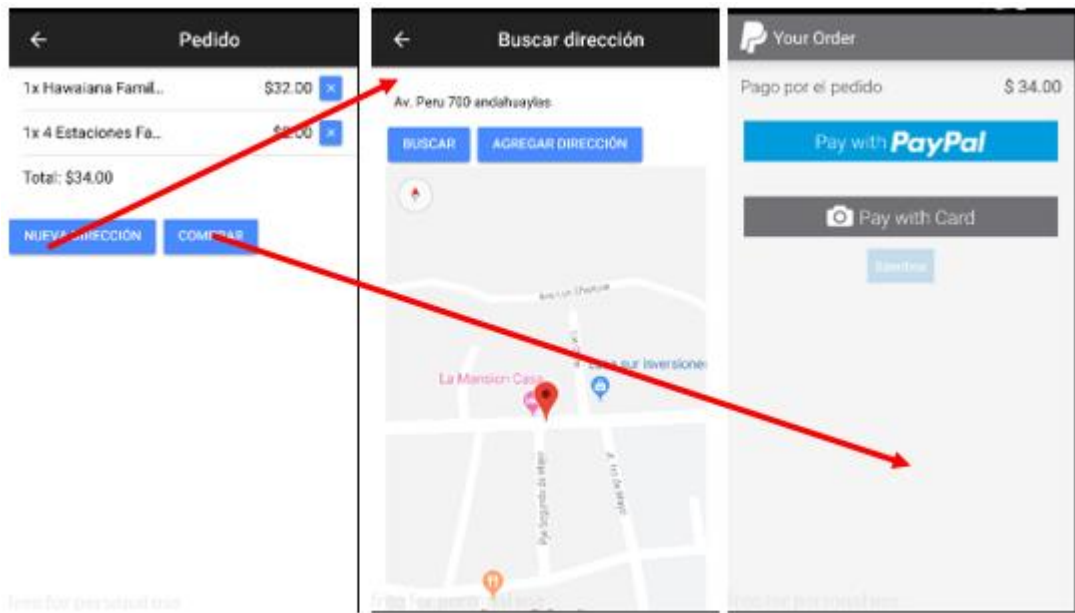


Figura N° 52: Generar Pedido

Fuente: Elaboración Propia

- Luego de haber seleccionado la opción “Pay with PayPal” el usuario tendrá que ingresar a su cuenta de PayPal, para posteriormente generar el pedido de manera exitosa.

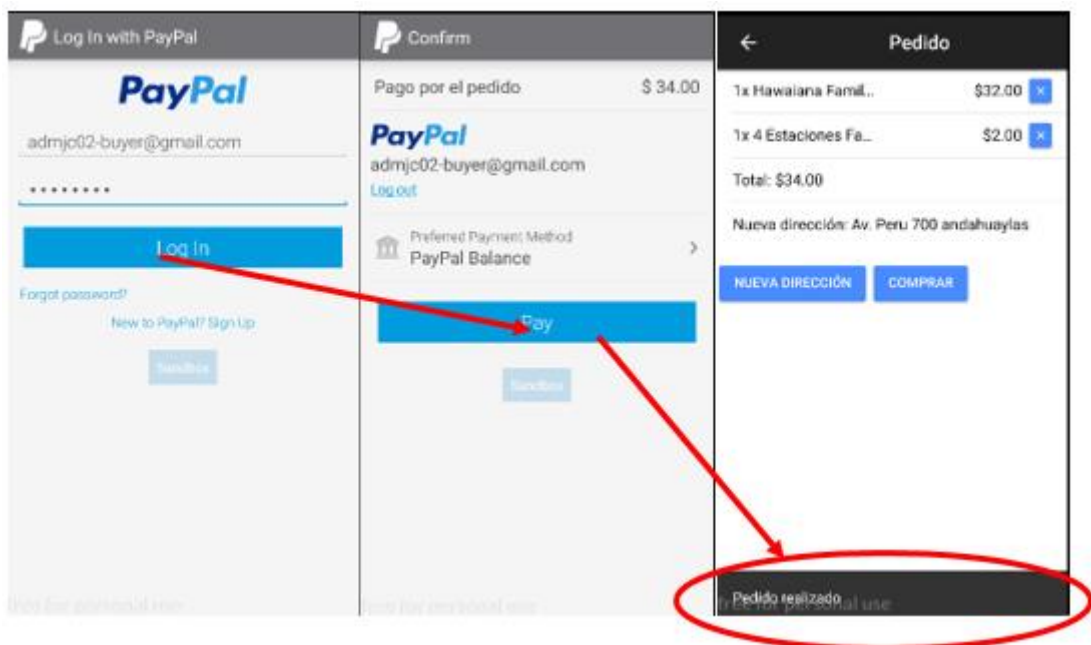


Figura N° 53: Realizar pago para generar el pedido.

Fuente: Elaboración Propia

MANUAL DE INSTALACIÓN Y CONFIGURACIÓN IONIC

A continuación, se da una demostración de la instalación y configuración del entorno de trabajo de IONIC.

- ✓ Descargar Node.js del siguiente link de acuerdo al tipo de Sistema Operativo:

<https://nodejs.org/es/>



Figura N° 54: Web descarga Node.js

Fuente: Portal Node.js

- ✓ Una vez descargado procedemos a instalarlo:

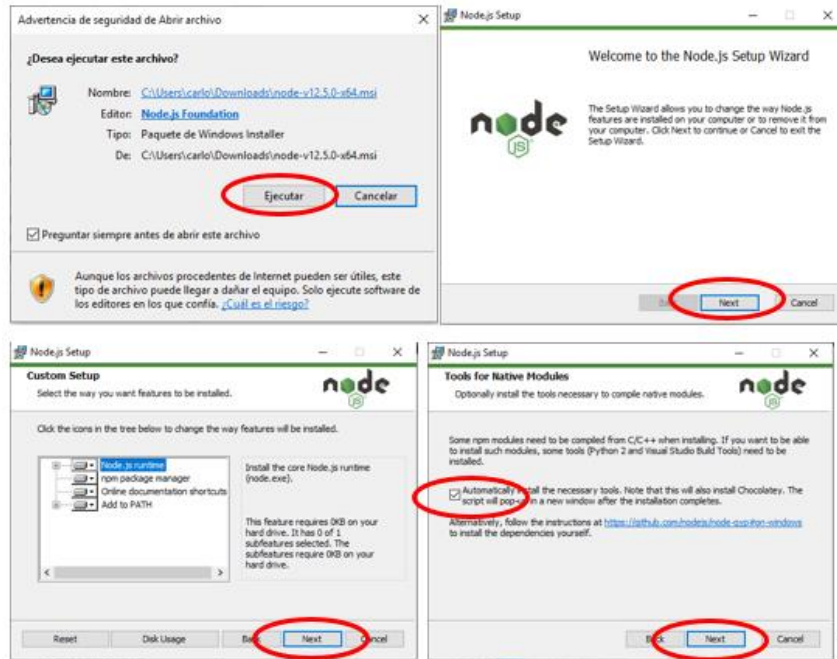


Figura N° 55: Pasos instalación Node.js

Fuente: Elaboración propia

- ✓ Como paso final daremos click en install.

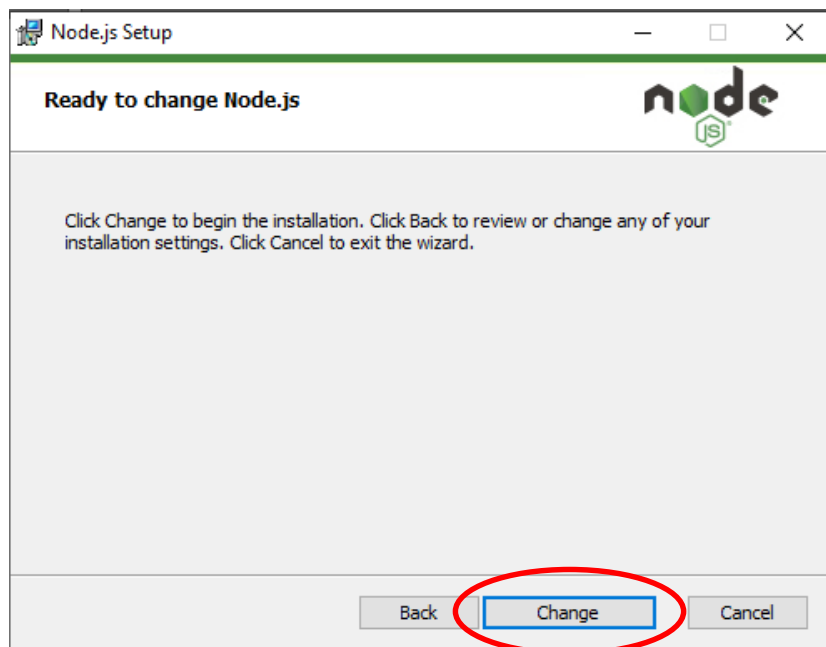


Figura N° 56: Instalación Node.js

Fuente: Elaboración propia

- ✓ Luego de haber instalado el Node.js procederemos a descargar e instalar el Visual Studio Code del siguiente link:

<https://code.visualstudio.com/download>

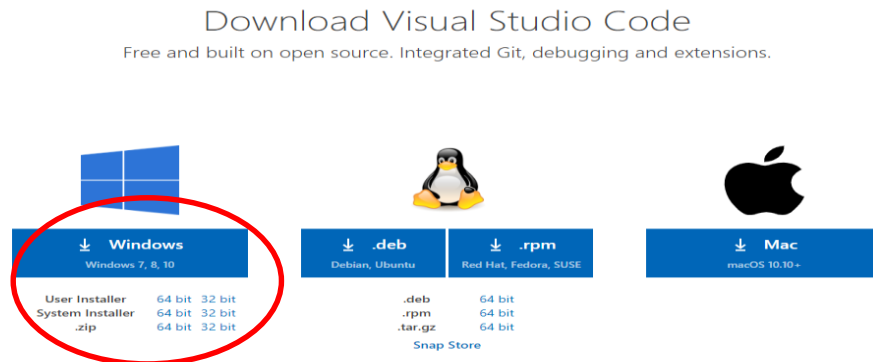


Figura N° 57: Portal de descarga Visual Code Studio

Fuente: Portal Visual Code Studio

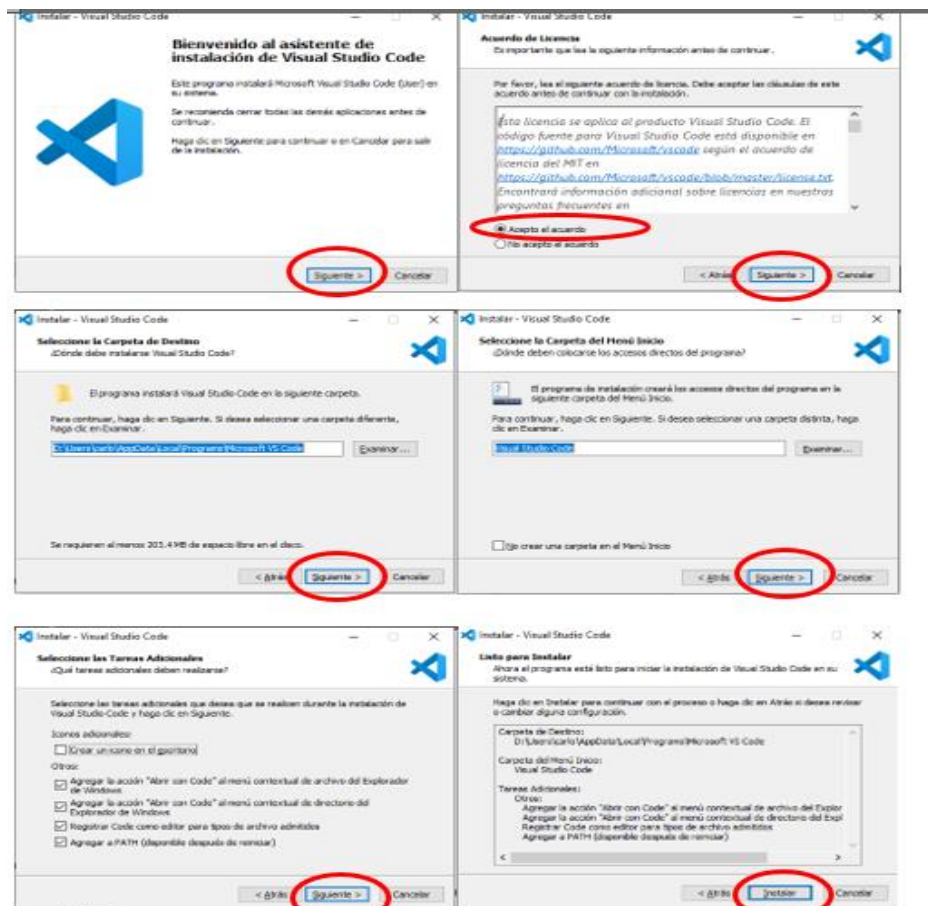


Figura N° 58: Pasos Instalación Visual Code Studio

Fuente: Elaboración propia

- ✓ Una vez finalizado la instalación del Visual Code Studio Procedemos a ejecutar el Visual Code Studio.



Figura N° 59: Acceso directo Visual Code Studio

Fuente: Elaboración Propia

- ✓ Una vez ejecutado el software y estando en el menú principal procedemos a ejecutar el terminal con el siguiente comando. Ctrl+ñ

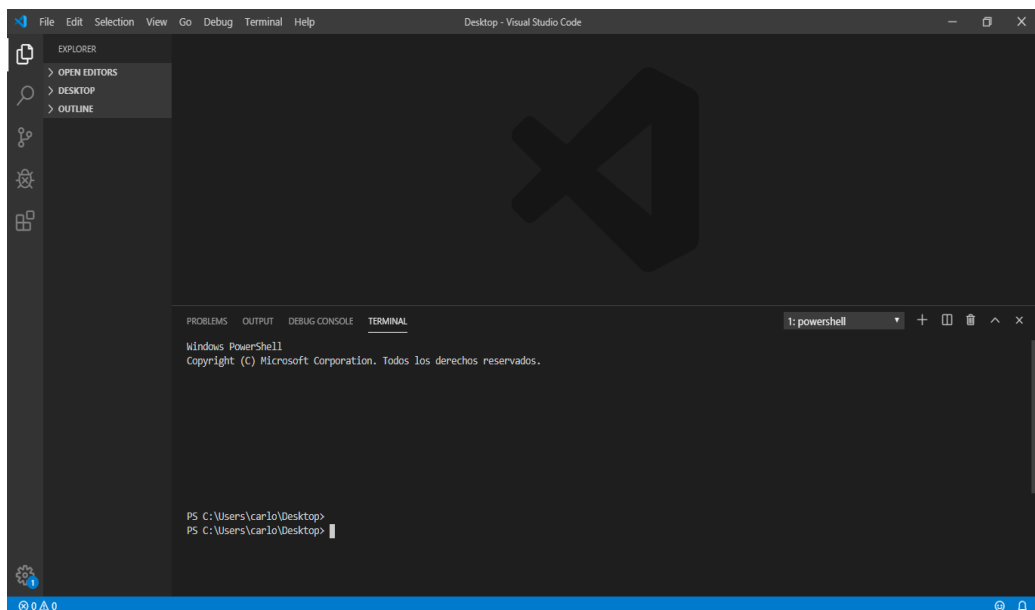


Figura N° 60: Menú principal Visual Code Studio

Fuente: Elaboración Propia

- ✓ Ahora procederemos a verificar la versión del Node.js con el comando:
node -v

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\carlo\Desktop>
PS C:\Users\carlo\Desktop> node -v
v12.5.0
PS C:\Users\carlo\Desktop> |
```

Figura N° 61: Verificar Versión del Node.js

Fuente: Elaboración propia

- ✓ Ahora procederemos a instalar Cordova con el siguiente comando:
npm install -g cordova

```
PS C:\Users\carlo\Desktop> npm install -g cordova
[.....] / rollbackFailedOptional: verb npm-session 08b22f0e3f31a65d
```

Figura N° 62: Instalación de Cordova

Fuente: Elaboración Propia

- ✓ Como paso siguiente instalaremos IONIC con el siguiente comando:
npm install -g ionic

```
PS C:\Users\carlo\Desktop> npm install -g ionic
[.....] \ rollbackFailedOptional: verb npm-session 4f6b99b0f773b601
```

Figura N° 63: Instalación IONIC

Fuente: Elaboración propia

- ✓ Una vez instalado IONIC procedemos a crear nuestro proyecto con el siguiente comando:
ionic start myApp blank

```
PS C:\Users\carlo\Desktop> ionic start myApp blank
```

Figura N° 64: Crear Proyecto Ionic

Fuente: Elaboración propia