

TRABAJO DE GRADO

Al-instante App: Digitalización del Servicio de Domicilios

Corporación Universitaria Remington.

Facultad de Ingeniería.

Ingeniería de Sistemas.

Autor: Juan José Agudelo Vinasco

Tutor: Gloria Amparo Lora Patiño

Trabajo de grado

2026

Dedicatoria

A Dios, por ser mi guía constante, por iluminar mi camino en los momentos de incertidumbre y por darme la fuerza necesaria para nunca rendirme. Sin su presencia en mi vida, este logro no habría sido posible.

A mi familia, porque han sido mi base, mi impulso y mi refugio. Su amor incondicional, sus palabras de aliento y su confianza en mí han sido fundamentales para alcanzar esta meta.

A mis padres, con todo mi amor y respeto. Gracias por cada sacrificio silencioso, por enseñarme con el ejemplo el valor del trabajo honesto, la disciplina y la perseverancia. Este logro es también el reflejo de sus esfuerzos y de la educación que me brindaron con tanto amor.

A mis hermanos, quienes han sido inspiración y compañía en este camino. Su apoyo constante, su cariño y sus consejos han sido un motor que me empujó a seguir, incluso en los días más difíciles.

Y a ti, Valen, mi compañera, mi amiga y mi amor, gracias por creer en mí incluso cuando yo dudaba, por tu paciencia infinita y por acompañarme con ternura y firmeza. Este logro no solo es mío, también es tuyo, porque estuviste en cada paso, en cada desvelo y en cada momento de celebración y dificultad.

A todos ustedes, gracias por ser parte esencial de esta etapa de mi vida. Este logro lleva un pedacito de cada uno en él.

Tabla de Contenidos

Resumen	8
Introducción	10
Planteamiento del problema.	12
Justificación	14
Objetivos	15
Objetivo General	15
Objetivos Específicos	15
Marco teórico	17
Transformación Digital y Comercio Electrónico	17
Aplicaciones Móviles en la Industria de Domicilios.....	21
Desarrollo de Aplicaciones Móviles	25
Logística de Última Milla y Distribución en Regiones Rurales	29
Economía Digital y su Impacto en el Desarrollo Regional.....	32
Normativa y Regulaciones en Colombia	36
Estado del Arte	41
Metodología	44
Fase de Análisis.....	45
Fase de Diseño	58
Fase de Programación.....	94
Fase de Pruebas	113
Resultados	125
Discusión	128
Conclusiones	130
Referencias	131

Índice de Figuras

Figura 1. Comercio Electrónico en América Latina.....	19
Figura 2. Valor de Mercado.....	23
Figura 3. Resultados de la encuesta: “¿Te gustaría hacer tus pedidos desde una aplicación en lugar de por WhatsApp?”	46
Figura 4. Resultados de la encuesta: “¿Crees que una app sería más rápida y práctica que escribir por WhatsApp?”	47
Figura 5. Resultados de la encuesta: “¿Preferirías ver el menú y precios directamente en una app?”	48
Figura 6. Resultados de la encuesta: “¿Te gustaría recibir notificaciones del estado de tu pedido en una app?.....	48
Figura 7. Resultados de la encuesta: “¿Estarías dispuesto(a) a descargar una app para pedir a domicilio más fácilmente?”	49
Figura 8. Planificación del Proyecto.....	55
Figura 9. Diagrama de casos de uso general de la aplicación Al-Instante App.....	60
Figura 10. Diagrama de caso de uso extendido (verificación OTP).	63
Figura 11. Diagrama de caso de uso extendido (Registro de usuario).	65
Figura 12. Diagrama de caso de uso extendido (Realizar pedido).	67
Figura 13. Diagrama de caso de uso extendido (Asignación de repartidor).	69
Figura 14. Diagrama de caso de uso extendido (Actualizar estado del pedido).	71
Figura 15. Diagrama de caso de uso extendido (Seguimiento del pedido).	73
Figura 16. Inicio de Sesión, verificación OTP, registro de usuario y menú principal de negocios aliados.	76
Figura 17. Almacenamiento y control de los datos	77

Figura 18. Ejemplo de colección usuarios y colección pedido	78
Figura 19. Carrito de Compras, productos seleccionados, c onfirmación de pedido, confirmación de dirección y esperando asignación de repartidor.....	79
Figura 20. Seguimiento del estado del pedido en el usuario, actualizaciones en tiempo real.	80
Figura 21. Diagrama UML de colecciones y relaciones	88
Figura 22. Diagrama de Actividades - Proceso General del.....	90
Figura 23. Diagrama de Secuencia – Funcionamiento General.....	91
Figura 24. Mapa de navegación del módulo de usuario y repartidor.	93
Figura 25. Panel de funciones de autenticación en Firebase Console.....	96
Figura 26. Reglas de seguridad implementadas en Cloud Firestore.....	97
Figura 27. Vista del desarrollo en Visual Studio Code con ejecución en emulador Android	99
Figura 28. Documentación oficial del lenguaje de programación Dart.....	100
Figura 29. Logo oficial de Flutter.	102
Figura 30. Repositorio del proyecto Al-instante App en GitHub (rama principal y estructura de carpetas).	105
Figura 31. Historial de confirmaciones (commits) del repositorio en GitHub.	106
Figura 32. Configuración del entorno de desarrollo Flutter.	108
Figura 33. Imagen del árbol del proyecto exportado desde la terminal.....	110
Figura 34. Ejecución de comandos para la configuración y prueba del sistema en entorno Flutter.	112
Figura 35. Captura Login y OTP.	116
Figura 36. Captura Home Cliente, Inicio Repartidor, Panel Super Usuario.	117
Figura 37. Captura formulario de registro.....	118

Figura 38. Captura panel admin, lista de pedidos, repartidores y negocios.119

Figura 39. Captura carrito y confirmación.....121

Figura 40. Captura lista repartidor.....122

Figura 41. Capturas de cambio de estados.123

Figura 42. Captura antes/después, repartidor y usuario.....125

Índice de Tablas

Tabla 1. Cuadro Comparativo: Aplicaciones de Entrega a Domicilio 43

Tabla 2. Identificación de los usuarios del sistema 50

Tabla 3. Requerimientos Funcionales..... 52

Tabla 4. Requerimientos no funcionales 53

Tabla 5. Matriz de trazabilidad de requisitos..... 56

Tabla 6. Verificación OTP..... 63

Tabla 7. Registro de usuario..... 65

Tabla 8. Realizar pedido 67

Tabla 9. Asignación de repartidor 70

Tabla 10. Actualización del estado del pedido 72

Tabla 11. Seguimiento del pedido..... 73

Tabla 12. Tabla PK/FK – Colección users (Clientes)..... 82

Tabla 13. Tabla PK/FK – Colección repartidores..... 83

Tabla 14. Tabla PK/FK – Colección superusuarios 84

Tabla 15. Tabla PK/FK – Colección pedidos (Documento principal)..... 85

Tabla 16. Subtabla: pedidos. productos [] (Array de mapas)..... 87

<i>Tabla 17. Tabla metodológica basada en tus commits</i>	106
<i>Tabla 18. Matriz de Trazabilidad</i>	115
<i>Tabla 19. TP-01 – Autenticación con OTP (RF-01)</i>	116
<i>Tabla 20. TP-03 – Identificación de Rol (RF-01)</i>	117
<i>Tabla 21. TP-04 – Registro de Usuario (RF-02)</i>	118
<i>Tabla 22. TP-05A – Gestión Repartidores Admin (RF-03)</i>	119
<i>Tabla 23. TP-05B – Gestión Negocios Admin (RF-03)</i>	120
<i>Tabla 24. TP-06 – Realizar Pedido (RF-04)</i>	120
<i>Tabla 25. TP-07 – Asignación de Repartidor (RF-05)</i>	121
<i>Tabla 26. TP-08 / TP-09 / TP-10 – Actualización de Estado (RF-06)</i>	122
<i>Tabla 27. TP-11 – Método de Pago (RF-07)</i>	123
<i>Tabla 28. TP-12 – Seguimiento Pedido Cliente (RF-08)</i>	124
<i>Tabla 29. Incidencias Detectadas y Corregidas</i>	Error! Bookmark not defined.

Resumen

El proyecto **AI-Instante App** se plantea como una propuesta tecnológica orientada a mejorar la gestión del servicio de entregas a domicilio en el municipio de Palestina, ubicado en el departamento de Caldas. Desde su inicio en el año 2024, operando inicialmente mediante la plataforma WhatsApp, ha representado una alternativa eficiente para vincular a los comercios locales con los clientes y los repartidores. Con más de dos años de funcionamiento, el servicio se ha posicionado como la única iniciativa de este tipo en la zona, contando actualmente con más de veinte establecimientos aliados y un promedio aproximado de 25 pedidos diarios, cifra que puede variar según la temporada.

Ante la necesidad de optimizar la administración de los pedidos, surge el desarrollo de una aplicación móvil que permite automatizar diferentes procesos del servicio. Esta herramienta busca mejorar la asignación de repartidores, disminuir posibles errores en la gestión de los pedidos y optimizar los tiempos de entrega. La aplicación AI-Instante App incorporará funcionalidades como el seguimiento de pedidos por estados y el envío de notificaciones, permitiendo mantener informados a los clientes sobre el progreso de sus solicitudes. Asimismo, se contempla la generación de estadísticas relacionadas con las ventas de los comercios, así como la creación de oportunidades de empleo para los repartidores.

El impacto del proyecto no se limita únicamente a mejorar la experiencia del usuario, sino que también contribuye al fortalecimiento del comercio local y a la generación de nuevas oportunidades laborales para los repartidores. Adicionalmente, el modelo planteado posee características de escalabilidad que permitirán, en el futuro, ampliar el

servicio hacia zonas cercanas como Santagueda y el municipio vecino de Chinchiná, donde la demanda de servicios de entrega a domicilio continúa en aumento.

Palabras clave

Posicionamiento de marketing, estrategias de innovación, empleo, tecnologías de información, base de dato, impacto económico, impacto social

Introducción

El servicio de entregas a domicilio ha experimentado un crecimiento importante en diversas regiones del mundo, impulsado por los cambios en los hábitos de consumo y el desarrollo de las tecnologías digitales. En municipios pequeños como Palestina, Caldas, este tipo de servicio ha sido limitado debido a la falta de plataformas tecnológicas especializadas que permitan una gestión eficiente de los pedidos y una organización adecuada en la distribución.

Aplicaciones como Rappi, Didi Food o iFood operan bajo modelos de negocio basados en economías de escala y una alta concentración de usuarios, por lo que no prestan servicio en zonas rurales con baja densidad poblacional como el municipio de Palestina. Estas plataformas suelen priorizar ciudades grandes y medianas donde la demanda y la rentabilidad son mayores, dejando por fuera a numerosos municipios que, a pesar de tener una demanda creciente, no cumplen con sus criterios comerciales (González, 2023).

Desde el año 2023, Al Instante Domicilios ha funcionado en el municipio mediante WhatsApp, permitiendo a los usuarios realizar pedidos a diferentes comercios, los cuales posteriormente son asignados manualmente a los repartidores disponibles. Aunque este modelo ha permitido atender una necesidad local, también ha presentado limitaciones como errores en la asignación de pedidos, demoras en las entregas y falta de trazabilidad en los procesos logísticos, lo que evidencia la necesidad de una solución tecnológica adaptada al contexto rural.

En este contexto surge Al-Instante App, una aplicación móvil orientada a optimizar la gestión de domicilios mediante una plataforma organizada, accesible e intuitiva. Su implementación busca digitalizar procesos que actualmente se realizan de forma manual, mejorando la experiencia del usuario, fortaleciendo el comercio local y facilitando la organización del trabajo de los repartidores.

El uso cotidiano de dispositivos móviles respalda esta iniciativa de digitalización. Gironés (2013) señala que los teléfonos móviles, al estar constantemente al alcance de las personas, han asumido un papel similar al de las computadoras personales. Esto evidencia la pertinencia de implementar soluciones móviles para servicios de uso diario como los domicilios.

Además, un estudio de Kantar (2020) indica que el 82% de los colombianos que solicitan domicilios lo hacen mediante aplicaciones móviles. Este dato demuestra la creciente adopción de estas plataformas y la necesidad de modernizar este servicio en contextos rurales donde dicha transformación aún no se ha desarrollado. En este sentido, Al-Instante App busca ofrecer una solución cercana, eficiente y adaptada a las condiciones del municipio de Palestina.

Planteamiento del problema.

Palestina es un municipio colombiano ubicado en la zona centro-sur del departamento de Caldas, dentro de la región del Eje Cafetero (Gobernación de Caldas, 2025). Su casco urbano se encuentra aproximadamente a 1.630 metros sobre el nivel del mar, sobre una colina desde la cual se observa el paisaje cafetero que caracteriza la región (Gobernación de Caldas, 2025). Esta área urbana tiene una extensión cercana a 245,64 hectáreas y alberga una población aproximada de 6.848 habitantes (Gobernación de Caldas, 2025). Entre los proyectos más relevantes del municipio se destaca la construcción del Aeropuerto Internacional del Café, iniciativa que busca mejorar la conectividad aérea de la región y facilitar la comercialización de productos agrícolas (Alcaldía de Palestina, Caldas, 2025).

En municipios rurales de Colombia, como Palestina, Caldas, la ausencia de herramientas tecnológicas para gestionar servicios de domicilios continúa siendo una limitación para acceder a soluciones logísticas modernas. Plataformas como Rappi, DiDi Food o iFood operan bajo modelos de negocio basados en economías de escala, los cuales concentran sus operaciones en ciudades con mayor densidad poblacional y actividad comercial. Debido a este enfoque, municipios con menor población, como Palestina, quedan excluidos de este tipo de plataformas digitales (González, 2023).

De acuerdo con las proyecciones del Departamento Administrativo Nacional de Estadística (DANE) para el año 2025, la cabecera municipal de Palestina cuenta con aproximadamente 6.075 habitantes, lo que equivale al 37,7% de la población total del

municipio, estimada en 16.121 personas (DANE, 2024). Esta distribución evidencia que una gran parte de los habitantes reside en veredas y zonas rurales, donde el acceso a servicios logísticos y tecnológicos es aún más limitado.

En un contexto nacional, Colombia ha presentado un crecimiento significativo en el uso de plataformas digitales para servicios de movilidad y entrega de productos. Según datos de Alianza In, citados por Forbes, este sector cuenta con más de 645.000 cuentas de repartidores, 135.000 comercios registrados y cerca de 7,5 millones de usuarios activos (Forbes Staff, 2024). No obstante, municipios como Palestina aún no se han integrado plenamente a esta transformación digital.

Actualmente, en el municipio funciona Al Instante Domicilios, un servicio de entregas gestionado de manera manual a través de WhatsApp y llamadas telefónicas, que ha permitido a la comunidad realizar pedidos a domicilio de forma básica. Sin embargo, los resultados de las encuestas realizadas indican que los usuarios consideran más conveniente contar con una aplicación que organice y agilice este proceso. Asimismo, los comercios aliados obtienen beneficios al trabajar con este servicio, lo que evidencia la necesidad de evolucionar hacia una herramienta tecnológica que mejore la experiencia tanto de clientes como de negocios locales.

Justificación

En Palestina, Caldas, el servicio de domicilios se ha gestionado principalmente de forma manual mediante WhatsApp, sin contar con una plataforma tecnológica que permita optimizar los procesos logísticos. Esta situación ha generado diversas dificultades, entre ellas errores en la asignación de los repartidores, tiempos de entrega variables y limitaciones para garantizar una experiencia eficiente tanto para los clientes como para los comercios locales. Frente a este panorama surge AI-Instante App, una solución tecnológica diseñada para superar estas limitaciones mediante una operación digital que facilite la gestión de pedidos, la asignación de entregas y el seguimiento en tiempo real, adaptándose a las condiciones del entorno local. De acuerdo con un estudio de la Universidad de América (2023), la incorporación de tecnologías digitales en los procesos de entrega permite mejorar la eficiencia operativa, aumentando la productividad y la competitividad del servicio.

En cuanto al impacto social, la implementación de AI-Instante App representa una oportunidad para fortalecer la inclusión digital en comunidades que históricamente han tenido menor acceso a procesos de transformación tecnológica. Facilitar el acceso a servicios modernos de domicilios no solo mejora la calidad de vida de los habitantes de Palestina, sino que también promueve el uso de herramientas tecnológicas entre los distintos actores del entorno local. La posibilidad de realizar pedidos de manera sencilla, segura y con seguimiento en tiempo real, así como el uso de opciones de pago digitales, contribuye a ampliar el acceso a los beneficios de la digitalización y a disminuir las brechas existentes en este tipo de servicios. Como señalan Ramírez y Sepúlveda (2018), la inclusión

digital en zonas rurales se entiende como un proceso mediante el cual las tecnologías se acercan a las comunidades para mejorar sus condiciones de vida.

Objetivos

Objetivo General

Desarrollar una aplicación móvil orientada a optimizar la gestión del servicio de domicilios en el municipio de Palestina, Caldas, con el fin de mejorar la eficiencia en la gestión de pedidos, fortalecer la comunicación entre usuarios y comercios aliados, y contribuir al impulso de la economía local mediante una herramienta tecnológica accesible y adaptada a las necesidades de la comunidad.

Objetivos Específicos

Fase de Análisis

- Realizar una revisión del estado del arte y aplicar encuestas a la comunidad del municipio de Palestina para identificar las necesidades y requerimientos que orienten el desarrollo de la aplicación.

Fase de Diseño

- Diseñar la estructura funcional de la aplicación AI-Instante App mediante la definición de la arquitectura del sistema, la elaboración de diagramas y el desarrollo

de prototipos de interfaz que permitan visualizar y validar el funcionamiento de la aplicación.

Fase de Programación

- Desarrollar el código fuente de la aplicación y estructurar la base de datos aplicando buenas prácticas de desarrollo, junto con la elaboración de documentación técnica que facilite su mantenimiento y escalabilidad.

Fase de Pruebas

- Realizar pruebas piloto con usuarios de la comunidad para evaluar la usabilidad y el rendimiento de la aplicación, permitiendo validar su funcionamiento y realizar los ajustes necesarios antes de su implementación.

Marco teórico.

Transformación Digital y Comercio Electrónico

La transformación digital se entiende como el uso estratégico de tecnologías para innovar en procesos, modelos de negocio y servicios digitales. Su base se encuentra en la digitalización, proceso mediante el cual los datos y actividades tradicionales se convierten en formatos que pueden ser gestionados por sistemas automatizados, lo que permite mejorar la eficiencia y generar mayor valor en las organizaciones (OECD, CONPES 3975, 2019).

La incorporación de tecnologías digitales dentro de las organizaciones requiere capacitación constante, la implementación de medidas de seguridad y el desarrollo de modelos de negocio innovadores. En este contexto, las estrategias de transformación digital se plantean como planes orientados a integrar nuevas tecnologías sin afectar la información histórica ni la identidad corporativa de las empresas, permitiendo así su adaptación y competitividad en entornos dinámicos (Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia, 2025).

El concepto de e-business se utiliza para describir la aplicación de las tecnologías de la información y la comunicación en las diferentes actividades empresariales, mientras que el e-commerce se relaciona específicamente con las transacciones comerciales realizadas por medios digitales. Para que estas operaciones se consideren dentro del comercio electrónico, procesos como compras, pedidos, pagos o financiamiento deben realizarse a través de redes abiertas y no propietarias (Tobar, 2015).

De acuerdo con la definición establecida por la OCDE, el comercio electrónico comprende la compra y venta de bienes o servicios mediante Internet entre empresas, individuos, hogares, gobiernos u otras organizaciones. En este tipo de transacciones, aunque la solicitud del producto o servicio se realiza en línea, tanto el pago como la entrega pueden efectuarse dentro o fuera de plataformas digitales (OCDE, 2011).

La pandemia de COVID-19 aceleró significativamente la adopción de canales digitales debido a las medidas de distanciamiento social implementadas en distintos países. Como consecuencia, el comercio electrónico experimentó un crecimiento notable, adelantando en pocos meses un proceso de transformación digital que anteriormente avanzaba de manera más gradual (Naciones Unidas, Konrad-Adenauer-Stiftung e.V. & Banco Interamericano de Desarrollo, 2021).

1. Evolución del Comercio Electrónico en Colombia

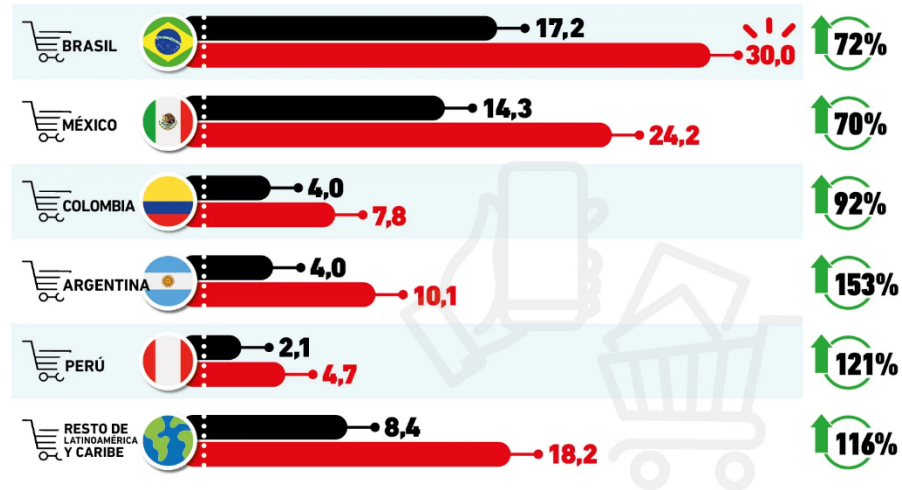
Colombia se posiciona como el tercer país con mayor desarrollo en comercio electrónico en América Latina, después de Brasil y México. Durante el año 2024, este sector alcanzó un valor aproximado de 52 mil millones de dólares y, según proyecciones, podría llegar a los 81 mil millones de dólares para el año 2027 si mantiene una tasa de crecimiento anual cercana al 16 % (Statista, 2024).

Figura 1. Comercio Electrónico en América Latina

EL BOOM DEL COMERCIO ELECTRÓNICO EN LATINOAMÉRICA

Valor de las ventas online en países de Latinoamérica en 2019 y 2024

CIFRAS EN MILES DE MILLONES DE DÓLARES.



Fuente: Statista, 2024

2. Tendencias actuales en el comercio digital.

Las tendencias actuales del comercio electrónico muestran que las plataformas de eCommerce deben ofrecer múltiples opciones de pago seguras y accesibles, con el fin de facilitar a los usuarios la realización de compras a través de distintos canales digitales. Esto implica garantizar procesos de pago ágiles y confiables en cada interacción con el cliente.

Según la Cámara Colombiana de Comercio Electrónico (CCCE), durante el último trimestre de 2023 se registraron aproximadamente 211,2 millones de transacciones digitales, lo que representa un crecimiento del 18,3 % en comparación con el mismo periodo de 2022. Asimismo, el uso de tarjetas de débito disminuyó al 47,8 %, mientras que

las transacciones con tarjetas de crédito aumentaron hasta alcanzar el 48,4 % (CCCE, 2024).

En Colombia, los medios de pago utilizados en el comercio electrónico se clasifican principalmente en tres categorías: débito a cuenta bancaria mediante PSE, tarjetas de crédito o débito y pagos en efectivo, ya sea en puntos de recaudo o contra entrega. Para el año 2024, la participación de estos medios fue del 58,5 % para débito a cuenta bancaria, 39,2 % para tarjetas y 2,4 % para pagos en efectivo. Estas cifras pueden presentar actualizaciones a medida que avanza el proceso de análisis de la información (PayU, Evertec / Place to Pay, PayValida, ePayco, Mercado Pago & Wompi, 2024).

3. Impacto del comercio digital y avances en Colombia

El comercio digital ha transformado la manera en que funcionan las economías, no solo a nivel global, sino también en regiones y municipios. El acceso a productos y servicios a través de internet se ha vuelto cada vez más sencillo, generando nuevas oportunidades para las personas y las empresas. Países como China y Estados Unidos lideran el desarrollo de este sector, mientras que economías emergentes como India y Brasil también han aprovechado estas herramientas digitales para impulsar el empleo y facilitar actividades de la vida cotidiana (Global Trade Culture, 2024).

En el caso de Colombia, desde enero de 2025 se implementó una medida relevante para fomentar el uso de servicios financieros digitales: Bancolombia y Davivienda eliminaron los costos asociados a las transferencias entre cuentas digitales. Esta decisión ha

favorecido el uso de medios electrónicos de pago al reducir las barreras económicas, permitiendo que más personas puedan participar en el sistema financiero (El País, 2025).

Otro cambio significativo se relaciona con la forma en que las empresas gestionan sus procesos de entrega. Cada vez más organizaciones incorporan tecnologías para administrar inventarios, realizar seguimiento a los pedidos y planificar de manera más eficiente sus operaciones logísticas. Estas herramientas han permitido mejorar la rapidez y la organización en las entregas, aspecto fundamental en un contexto donde el comercio por internet continúa en crecimiento (Analdex, 2021).

Aplicaciones Móviles en la Industria de Domicilios

El uso de aplicaciones móviles para solicitar domicilios ha aumentado considerablemente en Colombia, especialmente después de la pandemia. Muchas personas comenzaron a utilizar plataformas digitales para realizar sus pedidos, convirtiendo este tipo de servicios en parte de la rutina diaria. Un estudio realizado por Kantar señala que, en 2020, el 82 % de los colombianos que solicitaban domicilios lo hacían a través de aplicaciones móviles, lo que evidencia un cambio significativo en los hábitos de consumo (Kantar, 2020).

Sin embargo, los servicios de domicilios tienen antecedentes mucho más antiguos. Durante el siglo XX, con la expansión del uso del teléfono, numerosos restaurantes empezaron a recibir pedidos directamente mediante llamadas telefónicas. Aunque este

sistema permitió ampliar el servicio a domicilio, presentaba limitaciones relacionadas con la rapidez del proceso y la cobertura del servicio (McKinsey & Company, 2024).

La aparición de los teléfonos inteligentes y las aplicaciones móviles transformó completamente este modelo. A comienzos de la década de 2010 surgieron plataformas como Postmates (2011), DoorDash (2013) y Uber Eats (2014), las cuales lograron conectar de manera más eficiente a clientes, restaurantes y repartidores. Este desarrollo tecnológico marcó un cambio importante en la forma en que se solicitan alimentos y servicios de entrega.

Actualmente, el mercado global de domicilios está dominado por grandes plataformas digitales. Delivery Hero, por ejemplo, opera en más de 70 países y cuenta con cerca de 500.000 restaurantes afiliados. En China, empresas como Meituan y Ele.me lideran el sector, respaldadas por compañías tecnológicas de gran tamaño como Tencent y Alibaba (Statista, 2024).

1. Casos de éxito en Colombia.

En el año 2015 surgió en Bogotá la plataforma Rappi, una aplicación que rápidamente logró expandirse a distintos países de América Latina. Inicialmente enfocada en la entrega de alimentos, con el tiempo amplió su oferta para incluir el envío de productos farmacéuticos, retiro de dinero en efectivo y otros servicios a domicilio (Rappi, 2025).

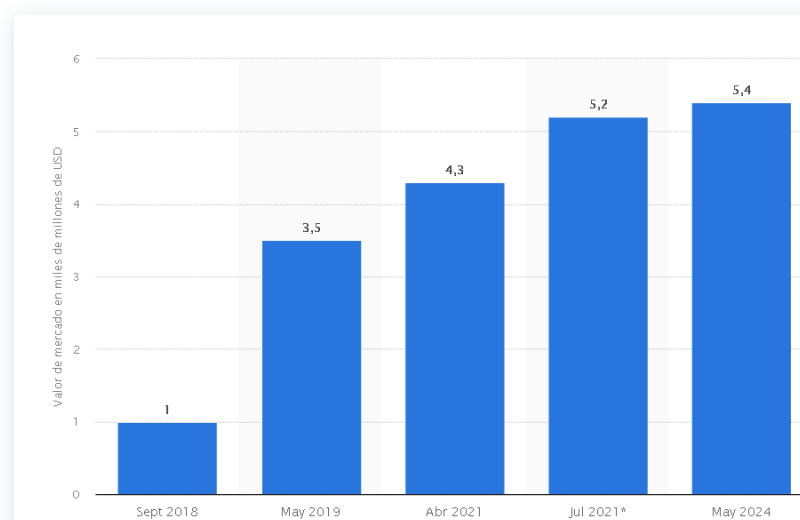
Para mayo de 2024, Rappi se consolidó como una de las startups más destacadas de Colombia, alcanzando una valoración de mercado cercana a los 5.400 millones de

dólares. Esta cifra representa un crecimiento aproximado del 440 % frente a septiembre de 2018, momento en el que la empresa superó los 1.000 millones de dólares y obtuvo la categoría de empresa unicornio (Statista, 2024).

Figura 2. Valor de Mercado

Valor de mercado de Rappi de 2018 a 2024

(en miles de millones de dólares estadounidenses)



Fuente: Statista

2. Modelos de Negocio en Apps de Domicilios

El servicio de domicilios en Colombia ha experimentado una evolución importante en los últimos años, impulsada principalmente por el avance de la tecnología y los cambios en los hábitos de consumo de los usuarios. Esta transformación ha modificado la manera en que empresas y consumidores interactúan dentro del mercado de entregas a domicilio (Revista La Barra, 2023).

3. Características Esenciales de una App de Domicilios

Una interfaz clara y fácil de usar contribuye a mejorar la experiencia del usuario, ya que permite navegar por la aplicación de forma sencilla, buscar productos con mayor rapidez y realizar pedidos sin complicaciones. Este aspecto resulta fundamental para mantener la satisfacción de los clientes y fortalecer su fidelidad hacia la plataforma (Quickworks, 2024).

Estas aplicaciones también permiten que los usuarios gestionen sus pedidos de manera eficiente, ofreciendo funciones como la programación de entregas y el seguimiento en tiempo real del estado del pedido. Al mismo tiempo, brindan a restaurantes y repartidores herramientas que facilitan la administración y supervisión de los pedidos durante todo el proceso (Redwerk, 2023).

Asimismo, la incorporación de distintos métodos de pago, como tarjetas de crédito o débito y billeteras digitales, permite realizar transacciones de forma segura y cómoda. Esta integración contribuye a aumentar la confianza de los usuarios en la plataforma y facilita la realización de compras en línea (Elluminati Inc., 2023).

4. Experiencia del usuario (UX/UI).

El éxito de una aplicación móvil de entrega de alimentos depende de su capacidad para innovar y ofrecer una experiencia de usuario superior. Para destacar en un mercado competitivo, es fundamental contar con un diseño intuitivo, opciones de personalización, programas de fidelización, integración con pagos en línea y un sistema eficiente de ubicación y seguimiento. Estas características mejoran la comodidad y eficiencia del

servicio, incentivando la lealtad del usuario y optimizando la gestión de pedidos y entregas (Food App, 2024).

Desarrollo de Aplicaciones Móviles

Una interfaz correctamente diseñada contribuye a mejorar la experiencia del usuario, ya que facilita la navegación dentro de la aplicación y favorece la permanencia de los usuarios en la plataforma. De acuerdo con Microsoft Power Apps, la calidad de la interfaz de usuario es un elemento clave para el funcionamiento adecuado de una aplicación móvil (Microsoft, 2023).

La velocidad de carga también representa un factor determinante para la satisfacción del usuario. Un tiempo de respuesta rápido permite una interacción más fluida con la aplicación y mejora la percepción del servicio ofrecido (Microsoft, 2023).

La seguridad de la información es otro aspecto fundamental en el desarrollo de aplicaciones móviles, ya que garantiza la protección de los datos de los usuarios y fortalece la confianza en el uso de la plataforma (Microsoft, 2023).

Finalmente, disponer de un sistema de atención o soporte al cliente eficiente contribuye a mejorar la experiencia general del usuario, brindando soluciones oportunas y generando mayor confianza en el uso de la aplicación (Microsoft, 2023).

1. Tecnologías para Desarrollo de Aplicaciones Móviles

Flutter:

Framework de código abierto creado por Google que permite desarrollar aplicaciones para diferentes plataformas a partir de una sola base de código. Se caracteriza por ofrecer un alto rendimiento y por facilitar la creación de interfaces de usuario personalizables y dinámicas (Bambú Mobile, 2024).

React Native:

Desarrollado por Meta (anteriormente Facebook), este framework permite construir aplicaciones móviles utilizando JavaScript y React. Facilita el desarrollo multiplataforma y cuenta con una amplia comunidad de soporte (Armadillo Amarillo, 2022).

Kotlin y Swift:

Lenguajes de programación recomendados para el desarrollo nativo en Android e iOS, respectivamente. Kotlin es promovido por Google para Android, mientras que Swift es el lenguaje oficial de Apple para iOS (Armadillo Amarillo, 2022).

Ionic:

Framework de código abierto que emplea tecnologías web como HTML, CSS y JavaScript para el desarrollo de aplicaciones móviles híbridas. Su principal

ventaja es que permite construir aplicaciones que pueden ejecutarse en distintas plataformas utilizando una única base de código (Redwerk, 2024).

2. Lenguajes de programación.

Flutter: ¿Qué es y cómo funciona?

Flutter es un kit de desarrollo de software (SDK) de código abierto desarrollado por Google que permite crear aplicaciones nativas de alta calidad para diversas plataformas, como Android, iOS, Web, Windows, macOS y Linux. Su funcionamiento se basa en una única base de código programada en el lenguaje Dart, lo que facilita el desarrollo multiplataforma (Flutter, 2025).

Dart SDK:

Lenguaje de programación orientado al desarrollo de interfaces de usuario, que permite la compilación anticipada (AOT) para optimizar el rendimiento en entornos de producción y la compilación en tiempo de ejecución (JIT) para facilitar las tareas de desarrollo y depuración (Flutter, 2025).

Flutter Engine:

Motor de renderizado desarrollado principalmente en C++ que utiliza la biblioteca gráfica Skia para representar los componentes de la interfaz de usuario. Este motor ofrece soporte para gráficos, animaciones, manejo de archivos, redes,

accesibilidad y otras funcionalidades necesarias para el funcionamiento de la aplicación (Flutter, 2025).

Foundation Library:

Conjunto de bibliotecas desarrolladas en el lenguaje Dart que ofrecen las clases y funciones necesarias para crear aplicaciones en Flutter. Estas bibliotecas incluyen APIs que permiten la comunicación entre la aplicación y el motor del framework (Flutter, 2025).

Widgets:

Flutter emplea un sistema basado en widgets para la construcción de la interfaz de usuario. Entre sus principales conjuntos se encuentran Material Design, que sigue las pautas de diseño establecidas por Google, y Cupertino, que reproduce el estilo visual característico del sistema iOS de Apple (Flutter, 2025).

3. Sistemas de bases de datos.

Firestore: ¿Qué es y cómo funciona?

Firestore es una plataforma de desarrollo de aplicaciones móviles y web creada por Google, que proporciona una variedad de herramientas y servicios para ayudar a los desarrolladores a construir aplicaciones de alta calidad. Entre sus servicios más destacados se encuentran sus soluciones de bases de datos en la nube: **Firestore Realtime Database y Cloud Firestore.**

Firestore Realtime Database

Es una base de datos NoSQL alojada en la nube que almacena datos en formato JSON y los sincroniza en tiempo real con todos los clientes conectados. Esto permite que múltiples usuarios compartan y actualicen datos simultáneamente, lo que es ideal para aplicaciones que requieren colaboración en tiempo real, como chats o aplicaciones de seguimiento en vivo. Además, ofrece soporte sin conexión, lo que significa que las aplicaciones pueden seguir funcionando y sincronizarse una vez que se restablezca la conexión a Internet (Firestore, 2025).

Cloud Firestore

Es la evolución de Realtime Database, ofreciendo una base de datos NoSQL más flexible y escalable. Organiza los datos en documentos y colecciones, lo que facilita consultas más complejas y estructuración de datos jerárquicos. Al igual que Realtime Database, proporciona sincronización en tiempo real y soporte sin conexión, pero con mejoras en la escalabilidad y rendimiento para aplicaciones más grandes y complejas (Firestore, 2025).

Logística de Última Milla y Distribución en Regiones Rurales

La logística de última milla en zonas rurales presenta diversos retos asociados a características propias del territorio, como la baja densidad de población, las limitaciones en la infraestructura vial y las grandes distancias entre los puntos de entrega. Estas condiciones incrementan los costos operativos y dificultan la eficiencia en la distribución

de productos. De acuerdo con Proships (2023), en las áreas rurales los costos de envío suelen ser mayores debido al reducido número de entregas y a las distancias que deben recorrer los repartidores.

En el caso de Colombia, el Observatorio de eCommerce del Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC) señala que las regiones rurales alejadas de las capitales departamentales presentan mayores tiempos de entrega y tarifas más elevadas en comparación con las ciudades principales. Esta situación está relacionada con la falta de infraestructura adecuada y con la dispersión geográfica de la población en estas zonas (MinTIC, 2021).

Con el fin de enfrentar estos desafíos, se han implementado diversas estrategias logísticas, entre ellas la creación de puntos de entrega comunitarios. Estos espacios permiten concentrar las entregas en ubicaciones estratégicas dentro de las comunidades rurales, facilitando el acceso de los habitantes a los productos y contribuyendo a la reducción de los costos logísticos (Hedyla, 2023).

De igual manera, algunas empresas han invertido en ampliar sus redes de distribución en territorios rurales. Un ejemplo de ello es Amazon, que ha fortalecido su sistema de entregas en zonas rurales de Estados Unidos mediante la creación de nuevas estaciones logísticas y la vinculación de conductores locales, con el objetivo de mejorar los tiempos de envío y atender la demanda en pequeñas ciudades y comunidades alejadas (Reuters, 2025).

1. Retos Logísticos en Zonas Rurales y Municipios.

El deterioro de la infraestructura vial y la falta de mantenimiento adecuado dificultan el acceso a numerosas zonas rurales, lo que incrementa tanto los tiempos de desplazamiento como los costos asociados al transporte. De acuerdo con la Encuesta Nacional Logística 2020, el 27,7 % de las empresas señaló el mal estado de las vías como uno de los principales problemas que afectan sus operaciones logísticas.

Otro desafío importante es el limitado acceso a las tecnologías de la información y la comunicación (TIC). La falta de conectividad y de herramientas tecnológicas en estas zonas dificulta la implementación de soluciones logísticas modernas, como el seguimiento de pedidos en tiempo real y la optimización de rutas de entrega.

2. Infraestructura vial y conectividad.

La infraestructura vial y la conectividad en las zonas rurales de Colombia presentan diversas dificultades que influyen directamente en el desarrollo económico y social de estas regiones. Las condiciones de las vías y las limitaciones en la conectividad afectan la movilidad de las personas y el transporte de bienes, especialmente en territorios alejados de los centros urbanos.

De acuerdo con la Contraloría General de la República (2023), únicamente el 29 % de las vías terciarias del país se encuentra en buen estado, mientras que el 71 % restante presenta condiciones regulares, malas o incluso impracticables. Esta situación genera dificultades para el transporte de productos agrícolas, limita el acceso a servicios básicos y afecta la calidad de vida de millones de habitantes en las zonas rurales.

Con el propósito de mejorar esta situación, el Gobierno Nacional implementó el Plan Nacional de Vías para la Integración Territorial (PNVIT) 2022-2026. Este programa contempla una inversión superior a los 15 billones de pesos colombianos destinada a la construcción, pavimentación y mantenimiento de más de 20.000 kilómetros de vías terciarias durante los próximos años, con especial énfasis en las regiones rurales más apartadas del país.

No obstante, a pesar de estas iniciativas, aún persisten importantes desafíos en la ejecución de proyectos de infraestructura vial y conectividad en áreas rurales. Entre los principales obstáculos se encuentran la falta de recursos, problemas de corrupción y deficiencias en la calidad de algunas construcciones. Según un estudio del Fondo Monetario Internacional (FMI), recorrer 300 kilómetros en Colombia puede tomar cerca de siete horas, aproximadamente el doble del tiempo que en otros países, debido a estas dificultades en la infraestructura vial.

Economía Digital y su Impacto en el Desarrollo Regional

La economía digital se ha consolidado como un factor fundamental para el desarrollo regional, especialmente en países como Colombia, donde la incorporación de tecnologías digitales puede generar efectos importantes en la productividad y en los procesos de inclusión social.

Un estudio realizado para Google en conjunto con el Sistema Económico Latinoamericano y del Caribe señala que Colombia podría alcanzar un impacto económico cercano a los 114 mil millones de dólares para el año 2030 mediante una adopción amplia

de las oportunidades digitales. Esta cifra representaría aproximadamente el 22 % del producto interno bruto (PIB) del país (SELA, 2023).

Para enfrentar los desafíos asociados a esta transformación, el Banco Mundial propone diversas prioridades para Colombia, entre las cuales se destacan el fortalecimiento de la infraestructura digital, la modernización de los sistemas gubernamentales mediante plataformas públicas digitales y el desarrollo de una adecuada gobernanza de datos (Banco Mundial, 2023).

Asimismo, la Comisión Económica para América Latina y el Caribe (CEPAL) señala que la economía digital puede convertirse en un elemento clave para impulsar cambios estructurales en la región, promoviendo la inversión a largo plazo y la diversificación de las actividades productivas, lo que contribuye al crecimiento económico y a la inclusión social.

Además, la Comisión Económica para América Latina y el Caribe (CEPAL) enfatiza que la economía digital puede actuar como catalizador para el cambio estructural, fomentando la inversión a largo plazo y la diversificación de la estructura productiva, con impactos positivos en el crecimiento y la inclusión social.

1. Generación de Empleo a Través de Plataformas Digitales

Las plataformas digitales han emergido como una alternativa significativa para la generación de empleo, especialmente en contextos donde el acceso al mercado laboral tradicional es limitado. En Colombia, estas plataformas han proporcionado oportunidades laborales a diversos grupos, incluyendo migrantes y personas excluidas del empleo formal. Durante la pandemia de COVID-19, se convirtieron en una fuente crucial de ingresos para

muchos trabajadores que enfrentaban dificultades económicas (Maldonado & Vargas, 2024).

Según un estudio de la Fundación para la Educación Superior y el Desarrollo (Fedesarrollo), las plataformas digitales ofrecen beneficios como la flexibilidad horaria, la posibilidad de complementar ingresos y el acceso al mercado laboral para quienes enfrentan barreras en el empleo tradicional. Sin embargo, también presentan desafíos relacionados con la formalización laboral y la protección social de los trabajadores (Fedesarrollo, 2021).

A nivel regional, la Comisión Económica para América Latina y el Caribe (CEPAL) destaca que el trabajo intermediado por plataformas puede reforzar la informalidad laboral si no se implementan regulaciones adecuadas. No obstante, también reconoce que estas plataformas pueden facilitar la inclusión laboral de sectores tradicionalmente excluidos, siempre que se promuevan políticas que garanticen condiciones laborales justas (CEPAL, 2021).

2. Impacto de las apps de domicilios en el empleo.

Las plataformas digitales se han convertido en una alternativa relevante para la generación de empleo, especialmente en contextos donde el acceso al mercado laboral tradicional es limitado. En Colombia, estas plataformas han brindado oportunidades de trabajo a distintos grupos de la población, incluyendo migrantes y personas que enfrentan dificultades para acceder a empleos formales. Durante la pandemia de COVID-19, este tipo

de servicios representó una fuente importante de ingresos para muchos trabajadores afectados por la crisis económica (Maldonado & Vargas, 2024).

De acuerdo con un estudio realizado por la Fundación para la Educación Superior y el Desarrollo (Fedesarrollo), las plataformas digitales ofrecen ventajas como la flexibilidad en los horarios, la posibilidad de generar ingresos adicionales y el acceso al mercado laboral para personas que encuentran barreras en los esquemas de empleo tradicionales. No obstante, también se identifican desafíos relacionados con la formalización del trabajo y la garantía de protección social para quienes participan en estas actividades (Fedesarrollo, 2021).

A nivel regional, la Comisión Económica para América Latina y el Caribe (CEPAL) advierte que el trabajo mediado por plataformas digitales puede contribuir al aumento de la informalidad si no se establecen regulaciones adecuadas. Sin embargo, también reconoce que estas herramientas pueden favorecer la inclusión laboral de sectores tradicionalmente excluidos, siempre que se implementen políticas que aseguren condiciones laborales justas y adecuadas (CEPAL, 2021).

3. Formalización del trabajo de repartidores.

En Colombia, la formalización del trabajo realizado por repartidores de plataformas digitales ha sido un tema de discusión en distintos espacios académicos y legislativos durante los últimos años. Las condiciones laborales de estos trabajadores, conocidos comúnmente como “rappitenderos”, suelen caracterizarse por la ausencia de una relación laboral formal, la falta de prestaciones sociales y situaciones de trabajo que pueden considerarse precarias.

Un estudio desarrollado por la Universidad Pontificia Bolivariana analizó las posibles implicaciones que tendría la formalización laboral de los repartidores dentro del modelo de negocio de Rappi en Colombia, tomando como referencia experiencias regulatorias aplicadas a plataformas digitales en Europa. Los resultados del análisis indican que una regulación del trabajo en plataformas digitales no afectaría de manera significativa el modelo de negocio de la empresa, ya que las propuestas legislativas presentadas en el país buscan principalmente que la compañía asuma mayores responsabilidades frente a los riesgos asociados al trabajo. Asimismo, el estudio señala que, frente a cambios en el entorno normativo, la empresa podría implementar estrategias de adaptación como la inversión en innovación tecnológica, la atracción de talento humano, la entrega de elementos de protección y la capacitación de los repartidores en normas de seguridad vial (Universidad Pontificia Bolivariana, 2020).

Normativa y Regulaciones en Colombia

En Colombia, el comercio electrónico no se encuentra regulado por una única norma general, sino por un conjunto de disposiciones legales que se aplican según el tipo de actividad desarrollada dentro del entorno digital. Esto implica que aspectos como la protección al consumidor, la seguridad de las transacciones, el tratamiento de datos personales y el uso de medios de pago electrónicos están regulados por diferentes leyes específicas, lo que puede hacer que el marco jurídico resulte complejo para quienes inician operaciones en este sector (Observatorio de Comercio Electrónico, 2021).

Uno de los principales fundamentos legales del comercio electrónico en el país es la Ley 527 de 1999, considerada la primera normativa que estableció las bases jurídicas para las transacciones digitales en Colombia. Esta ley reconoce los mensajes de datos como equivalentes funcionales de los documentos físicos, otorgándoles validez legal. Asimismo, regula el uso de las firmas digitales y establece la figura de las entidades de certificación, encargadas de garantizar la autenticidad y seguridad de estos procesos (Congreso de la República de Colombia, 1999).

Con el paso del tiempo, se han incorporado otras normativas relevantes para fortalecer el marco legal del comercio digital. Entre ellas se encuentra la Ley 1581 de 2012, que regula el tratamiento de los datos personales, un aspecto fundamental debido al volumen de información que se intercambia entre empresas, plataformas y usuarios en los entornos digitales. De igual forma, la Ley 1266 de 2008 aborda el habeas data y la gestión de información financiera y crediticia.

En materia de protección al consumidor, la Ley 1480 de 2011, conocida como el Estatuto del Consumidor, establece los derechos y responsabilidades tanto de compradores como de vendedores en entornos digitales. Esta normativa exige que los comercios informen de manera clara las condiciones de los productos o servicios ofrecidos y garantiza mecanismos para la devolución del dinero o el cambio del producto en caso de inconformidad.

Más recientemente, entidades como la Superintendencia de Industria y Comercio (SIC) han publicado guías y circulares con el objetivo de orientar a las empresas sobre el cumplimiento de estas disposiciones en entornos digitales. Además, esta entidad ha

fortalecido su función de vigilancia y control frente al crecimiento acelerado del comercio electrónico en el país durante los últimos años.

1. Legislación sobre Plataformas de Domicilios

En Colombia, la regulación de las plataformas de domicilios ha tenido avances en los últimos años debido al crecimiento de este tipo de servicios y a la necesidad de garantizar mejores condiciones para quienes trabajan en ellos. En 2024, la Cámara de Representantes aprobó varios artículos dentro de la reforma laboral impulsada por el gobierno nacional, en los cuales se establece que los trabajadores vinculados a plataformas digitales como Rappi, Uber Eats o DiDi Food deben tener claridad sobre su tipo de relación laboral. Esta puede darse como trabajadores dependientes, mediante un contrato laboral formal, o como trabajadores independientes con responsabilidades compartidas en materia de seguridad social. En ambos casos, el objetivo es fortalecer las condiciones laborales y la protección de los repartidores (El País, 2024).

Por otra parte, a nivel local se han implementado medidas regulatorias para este sector. La Alcaldía de Bogotá, por ejemplo, expidió el Decreto 082 de 2021, el cual establece lineamientos para las empresas de domicilios que operan en la ciudad. Entre las disposiciones se incluye la obligación de verificar los antecedentes judiciales de los repartidores, registrar las bicicletas en el sistema “Bici Bogotá”, portar información visible del vehículo utilizado y adoptar medidas que contribuyan a la seguridad y al orden en el espacio público (Alcaldía de Bogotá, 2021). Estas acciones representan avances hacia una regulación más estructurada del sector y pueden servir como referencia para otros municipios del país.

2. Protección de Datos y Seguridad Digital

El Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC) ha impulsado la Política de Seguridad Digital con el propósito de que ciudadanos y empresas comprendan los riesgos presentes en el entorno digital y adopten prácticas adecuadas para prevenir incidentes, proteger su información y responder ante posibles delitos informáticos. Esta política busca fortalecer una cultura de seguridad digital en los distintos sectores del país (MinTIC, 2025).

De igual forma, el Decreto 338 de 2022 estableció el Modelo de Gobernanza de Seguridad Digital, el cual define lineamientos, principios y procedimientos orientados a fortalecer la gestión de riesgos y la atención de incidentes en los entornos tecnológicos. Este modelo tiene como objetivo garantizar la confidencialidad, integridad y disponibilidad de la información en los sistemas utilizados por entidades públicas y privadas en Colombia (Presidencia de la República de Colombia, 2022).

3. Ley de Habeas Data y seguridad de la información en apps.

En Colombia, la protección de la información personal está regulada por la Ley Estatutaria 1581 de 2012, conocida como Ley de Habeas Data. Esta normativa establece los derechos que tienen los ciudadanos sobre el tratamiento de sus datos por parte de entidades públicas y privadas, incluyendo la garantía de confidencialidad, integridad y el acceso a mecanismos que permitan actualizar o eliminar dicha información. Asimismo, obliga a las

organizaciones responsables del manejo de datos a implementar medidas de seguridad que prevengan filtraciones o usos indebidos de la información (Congreso de Colombia, 2012).

En el caso de las aplicaciones móviles, estas medidas adquieren mayor relevancia debido a que muchas de ellas recopilan datos sensibles como la ubicación del usuario, números de contacto o información relacionada con métodos de pago. De acuerdo con Fortinet, empresa especializada en ciberseguridad, la ausencia de mecanismos adecuados de protección en las aplicaciones puede representar riesgos tanto para los usuarios como para las organizaciones que las desarrollan, generando posibles pérdidas de información o afectaciones a su reputación. Por esta razón, incorporar la seguridad digital desde la etapa de diseño se considera una práctica fundamental para el desarrollo responsable de aplicaciones móviles (Fortinet, 2025).

4. Regulaciones sobre pagos electrónicos y cumplimiento normativo.

En Colombia, la regulación de los pagos electrónicos ha evolucionado con el objetivo de fomentar la inclusión financiera y fortalecer la seguridad en las transacciones digitales. La Ley 1735 de 2014 creó las Sociedades Especializadas en Depósitos y Pagos Electrónicos (SEDPE), las cuales permiten que entidades no bancarias ofrezcan servicios financieros básicos, como depósitos y transferencias, bajo la supervisión de la Superintendencia Financiera de Colombia. Esta normativa busca ampliar el acceso a servicios financieros formales para poblaciones que tradicionalmente han estado excluidas del sistema bancario (Congreso de Colombia, 2014).

Posteriormente, el Decreto 1297 de 2022 introdujo disposiciones relacionadas con el modelo de finanzas abiertas, incluyendo la actividad de iniciación de pagos a través del

Sistema de Pago de Bajo Valor. Además, esta regulación establece lineamientos sobre el tratamiento y la comercialización de los datos personales de los consumidores financieros por parte de las entidades supervisadas por la Superintendencia Financiera, así como la prestación de servicios dentro de ecosistemas digitales. Con ello se busca fortalecer el cumplimiento normativo en el entorno de los pagos electrónicos (Ministerio de Hacienda y Crédito Público, 2022).

Estado del Arte

El análisis de aplicaciones existentes permite ubicar el proyecto Al-Instante App dentro del contexto actual de soluciones digitales enfocadas en la entrega de productos a domicilio y en la optimización de la interacción entre clientes, comercios y repartidores. Plataformas como Rappi, iFood y PedidosYa evidencian la alta aceptación que han tenido este tipo de servicios en la región, ya que responden a la necesidad de rapidez y conveniencia al momento de adquirir productos.

Al-Instante App toma como referencia estos modelos de funcionamiento, pero adapta su propuesta a un entorno local mediante un enfoque comunitario orientado a fortalecer la economía del municipio, simplificar la gestión de pedidos y ofrecer oportunidades de empleo flexible para repartidores. En este sentido, el estudio de estas aplicaciones resulta fundamental para identificar buenas prácticas y diferenciar la propuesta del proyecto, aprovechando características ya consolidadas y ajustándolas a las necesidades específicas del contexto donde se implementará.

Rappi

Rappi es una aplicación colombiana de domicilios que se ha consolidado como una de las plataformas más reconocidas en América Latina. Su modelo de negocio integra diversos servicios dentro de una misma aplicación, permitiendo a los usuarios solicitar alimentos, realizar compras en supermercados, recibir medicamentos e incluso gestionar diferentes diligencias a través de los denominados “rappitenderos”. Esta plataforma se destaca por su expansión regional, el uso de sistemas de pago digitales y sus estrategias de fidelización mediante membresías (Rappi, 2024).

iFood

Otra plataforma relevante en el mercado es iFood, la cual conecta restaurantes y comercios con usuarios que desean solicitar alimentos y otros productos desde sus dispositivos móviles. Su funcionamiento se basa en un modelo similar al de otras aplicaciones del sector, ofreciendo un catálogo de establecimientos asociados, diversas opciones de pago —ya sea en línea o contra entrega— y la posibilidad de realizar seguimiento en tiempo real del estado del pedido. La sencillez de su interfaz y su amplia cobertura han contribuido a que continúe siendo una de las aplicaciones más utilizadas para solicitar domicilios (iFood.com, 2024).

PedidosYa

PedidosYa es una plataforma ampliamente utilizada en distintos países de América Latina, especializada en la gestión de entregas rápidas de alimentos, bebidas y productos de conveniencia. Esta aplicación integra herramientas como la geolocalización de repartidores, métodos de pago seguros y promociones dirigidas a los usuarios. Su popularidad se debe principalmente a su facilidad de uso, la variedad de establecimientos disponibles y la red de repartidores independientes que permiten agilizar el proceso de entrega (PedidosYa, 2024).

Tabla 1. Cuadro Comparativo: Aplicaciones de Entrega a Domicilio

Criterio	Rappi	iFood	PedidosYa	Al-instante App
Cobertura geográfica	Ciudades principales y expansión internacional	Ciudades medianas y grandes en Colombia	Presencia en capitales y principales ciudades de LATAM	Municipios pequeños y zonas con baja densidad poblacional
Público objetivo	Público general, usuarios urbanos con alta conectividad	Usuarios en zonas urbanas con acceso a múltiples restaurantes	Usuarios en grandes ciudades, variedad de comercios	Comunidades locales, vecinos de barrios o pueblos
Modelo de repartidores	Repartidores independientes, alta rotación	Repartidores contratados por terceros o por comisión	Red de repartidores propios o asociados	2 a 3 repartidores locales, jóvenes de la misma zona
Modelo de confianza	Confianza por marca y procesos de seguridad de pago	Confianza por trayectoria y soporte técnico	Confianza basada en reputación regional	Modelo de cercanía: repartidores y negocios conocidos en la comunidad
Servicios adicionales	Multiservicio: pedidos, diligencias, supermercado, fintech	Principalmente comida a domicilio	Comida, bebidas, farmacias, conveniencia	Principalmente pedidos de alimentos y productos locales, con enfoque comunitario

Criterio	Rappi	iFood	PedidosYa	Al-instante App
Diferenciador clave	Amplia variedad de servicios, membresías premium	Amplia cobertura de restaurantes aliados	Velocidad de entrega y promociones	Accesibilidad para municipios no cubiertos por grandes apps, impulso a la economía local

Nota. * Elaboración propia basada en información de los sitios web oficiales de cada aplicación.

Metodología

El desarrollo y análisis de Al-Instante App se realizó de manera estructurada, progresiva y organizada por fases. Si bien no se adoptó formalmente una metodología ágil como Scrum, el proceso de construcción del sistema se orientó de forma natural hacia un modelo de desarrollo por prototipos. Este enfoque permitió construir versiones funcionales tempranas de la aplicación, facilitando la validación de los requerimientos, la identificación de mejoras y la incorporación de ajustes continuos a partir de pruebas reales.

El uso del modelo de prototipos resultó especialmente adecuado para el contexto del proyecto, ya que permitió adaptar el desarrollo a las necesidades reales de los usuarios finales. A través de iteraciones sucesivas, se evaluaron funcionalidades clave como la autenticación por número telefónico, la gestión de pedidos, la asignación de repartidores y el seguimiento del estado de las entregas, permitiendo refinar progresivamente el sistema hasta alcanzar una versión estable y funcional.

Fase de Análisis

En esta fase se recolectó y organizó información clave relacionada con el funcionamiento esperado de AI-Instante App. Para ello, se analizaron las necesidades de usuarios potenciales, considerando sus hábitos de consumo, la forma en que solicitan servicios de domicilios y las dificultades presentes en los procesos tradicionales de pedido y entrega.

A partir de este análisis se definieron los distintos perfiles de usuario del sistema, identificando sus roles y niveles de interacción dentro de la plataforma, tales como clientes, repartidores y administradores. De manera paralela, se llevó a cabo el levantamiento de requerimientos, en el cual se identificaron las funcionalidades principales que debía ofrecer la aplicación, como el registro y autenticación mediante OTP, la realización y seguimiento de pedidos, y la actualización de estados por parte del repartidor. Asimismo, se establecieron requisitos no funcionales relacionados con la seguridad de la información, la facilidad de uso y el rendimiento del sistema.

Esta fase permitió delimitar con claridad el alcance inicial del proyecto y sentó las bases para las etapas posteriores de diseño, desarrollo e implementación de la aplicación.

Aplicación de encuesta a usuarios

Como parte del proceso de análisis de necesidades, se diseñó y aplicó una encuesta breve dirigida a 30 usuarios que utilizan con frecuencia el servicio de domicilios. El propósito fue conocer su disposición frente al uso de una aplicación móvil para realizar pedidos. La encuesta estuvo compuesta por cinco preguntas cerradas, orientadas a

identificar el interés de los participantes en una alternativa digital, las ventajas percibidas frente al uso actual de WhatsApp y el nivel de aceptación de funcionalidades como menús visibles, notificaciones y facilidad de navegación.

A continuación, se presentan las preguntas que hicieron parte de la encuesta aplicada:

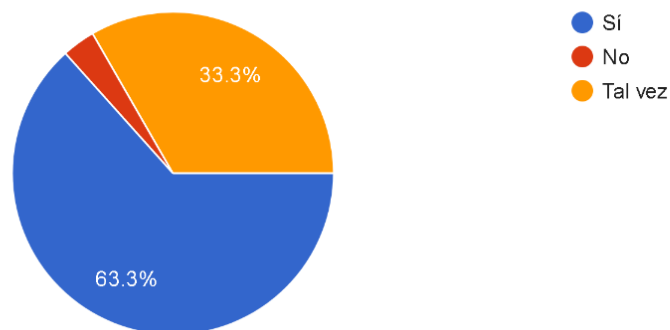
Encuesta aplicada a los 30 usuarios de servicios de domicilio

1. ¿Te gustaría hacer tus pedidos desde una aplicación en lugar de por WhatsApp?
2. ¿Crees que una app sería más rápida y práctica que escribir por WhatsApp?
3. ¿Preferirías ver el menú y precios directamente en una app?
4. ¿Te gustaría recibir notificaciones del estado de tu pedido en una app?
5. ¿Estarías dispuesto(a) a descargar una app para pedir a domicilio más fácilmente?

Figura 3. Resultados de la encuesta: “¿Te gustaría hacer tus pedidos desde una aplicación en lugar de por WhatsApp?”

¿Te gustaría hacer tus pedidos desde una aplicación en lugar de por WhatsApp?

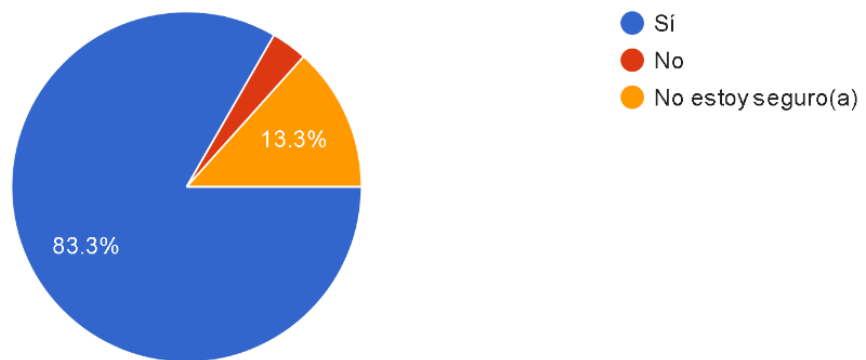
30 respuestas



Fuente: Elaboración Propia.

Figura 4. Resultados de la encuesta: “¿Crees que una app sería más rápida y práctica que escribir por WhatsApp?”

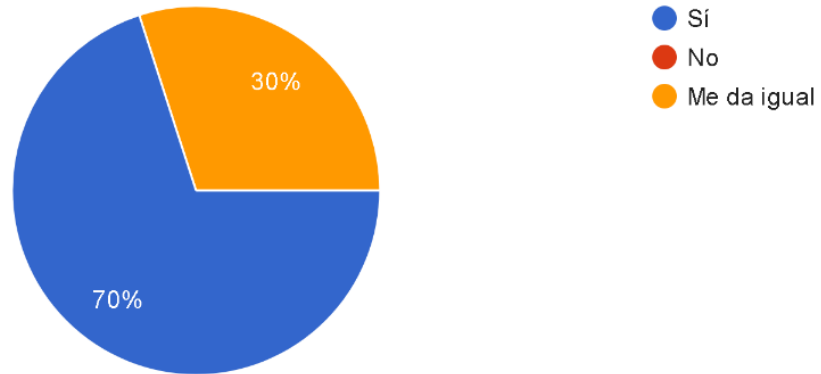
¿Crees que una app sería más rápida y práctica que escribir por WhatsApp?
30 respuestas



Fuente: Elaboración Propia

Figura 5. Resultados de la encuesta: “¿Preferirías ver el menú y precios directamente en una app?”

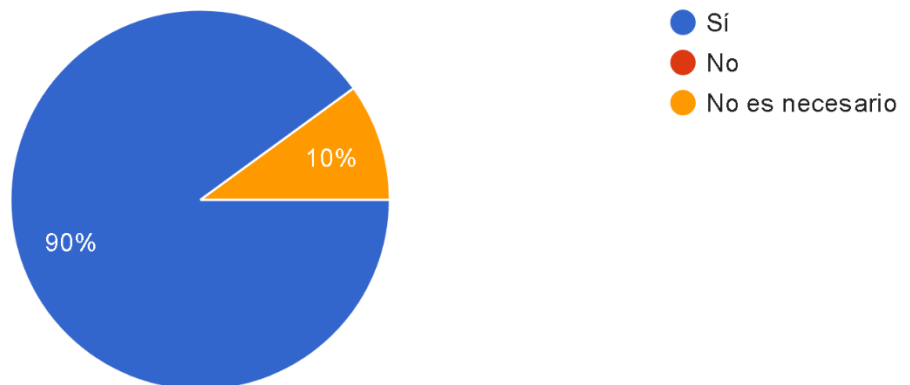
¿Preferirías ver el menú y precios directamente en una app?
30 respuestas



Fuente: Elaboración Propia

Figura 6. Resultados de la encuesta: “¿Te gustaría recibir notificaciones del estado de tu pedido en una app?”

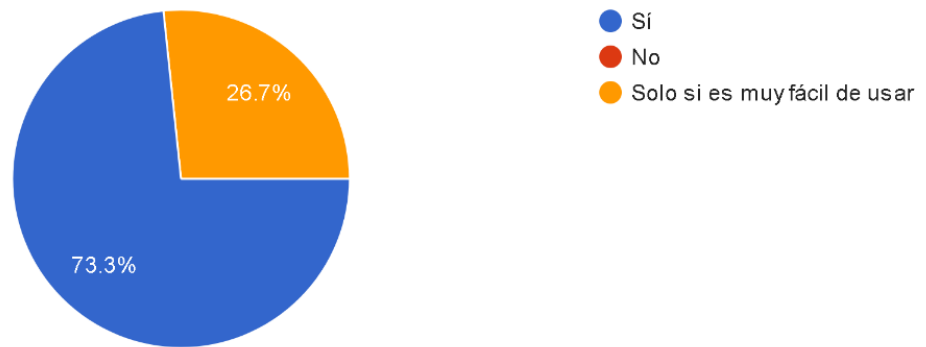
¿Te gustaría recibir notificaciones del estado de tu pedido en una app?
30 respuestas



Fuente: Elaboración Propia

Figura 7. Resultados de la encuesta: “¿Estarías dispuesto(a) a descargar una app para pedir a domicilio más fácilmente?”

¿Estarías dispuesto(a) a descargar una app para pedir a domicilio más fácilmente?
30 respuestas



Fuente: Elaboración Propia

Los resultados obtenidos a partir de la encuesta realizada a 30 participantes evidencian una alta aceptación frente a la implementación de **Al-Instante App** como una herramienta digital orientada a mejorar el servicio de domicilios. Las respuestas positivas de los encuestados respaldan la viabilidad del proyecto y reflejan el interés de la comunidad por contar con una solución que contribuya a optimizar la eficiencia del servicio y mejorar la experiencia de los usuarios.

Listado de usuarios del sistema

El sistema Al-Instante App está diseñado para atender las necesidades de tres tipos de usuarios principales, cada uno con un rol claramente definido dentro de la plataforma. Estos usuarios interactúan entre sí para garantizar el correcto funcionamiento del servicio

de pedidos y domicilios, integrando de manera eficiente a clientes, repartidores y administradores del sistema.

La identificación de los usuarios permite comprender sus responsabilidades, nivel de acceso y necesidades específicas, lo cual es fundamental para el levantamiento de los requisitos funcionales y el diseño de las funcionalidades de la aplicación.

Tabla 2. Identificación de los usuarios del sistema

Tipo de Usuario	Perfil del Usuario	Necesidades Principales
Cliente	Persona de la comunidad que utiliza la aplicación para explorar los comercios disponibles, realizar pedidos y hacer seguimiento al estado de sus solicitudes.	Realizar pedidos de forma rápida y sencilla, consultar el estado del pedido en tiempo real, recibir notificaciones sobre el proceso de entrega y contar con una experiencia de uso clara y segura.
Repartidor	Persona encargada de aceptar los pedidos disponibles, recoger los productos en los comercios aliados y entregarlos al cliente final, manteniendo actualizada la información del pedido.	Visualizar pedidos disponibles, aceptar entregas, actualizar los estados del pedido (preparando, en ruta y entregado) y contar con acceso a la información necesaria para completar la entrega.
Administrador	Usuario con permisos de gestión general del sistema, encargado de supervisar el funcionamiento de la plataforma y la operación de los actores involucrados.	Monitorear el estado de los pedidos, gestionar usuarios del sistema y supervisar la operación general de la aplicación.

Nota *Se identifican los tipos de usuarios del sistema junto con su perfil y necesidades principales, información clave para el levantamiento de los requisitos funcionales y el diseño de la aplicación AI-Instante App.

Levantamiento de requerimientos

La primera fase del proyecto correspondió al levantamiento de requerimientos, en la cual se identificaron las funcionalidades necesarias para los tres actores clave de la aplicación AI-Instante App: clientes, repartidores y administradores. Este proceso permitió definir de manera clara el alcance funcional del sistema, teniendo en cuenta el contexto real de uso y las necesidades operativas de la plataforma.

Los requerimientos funcionales identificados incluyen aspectos como la autenticación mediante número telefónico con código OTP, el registro de usuarios, la creación y seguimiento de pedidos, la gestión del carrito de compras, la visualización del estado del pedido en tiempo real y la actualización del estado por parte del repartidor durante el proceso de entrega. Asimismo, se contempla la interacción entre clientes y repartidores mediante información clave como la ubicación de entrega y los datos de contacto.

De forma complementaria, se definieron los requerimientos no funcionales, los cuales establecen criterios de calidad para el sistema. Entre estos se destacan la usabilidad, el rendimiento, la seguridad de la información, la escalabilidad del sistema, la portabilidad entre dispositivos móviles y la mantenibilidad del software. Estos requisitos garantizan que la aplicación no solo cumpla con sus funciones básicas, sino que también ofrezca una experiencia de uso adecuada y permita su evolución futura.

1. Requerimientos funcionales (RF)

Los requerimientos funcionales describen las características y comportamientos que debe cumplir la aplicación Al-Instante App para satisfacer las necesidades de los usuarios del sistema.

Tabla 3. Requerimientos Funcionales

ID del Requisito	Nombre del Requisito	Descripción del Requisito	Usuario
RF-01	Autenticación con OTP	El sistema permite el acceso mediante número de teléfono y verificación con código OTP, identificando el rol del usuario dentro de la aplicación (cliente, repartidor o administrador).	Cliente / Repartidor / Administrador
RF-02	Registro de Usuario	El sistema permite registrar usuarios clientes solicitando información básica como nombre, barrio, dirección y una descripción opcional para facilitar la entrega.	Cliente
RF-03	Gestión de Cuentas	El sistema permite al administrador visualizar y gestionar la información de los clientes y repartidores registrados en la plataforma.	Administrador
RF-04	Realizar Pedidos	El sistema permite a los clientes realizar pedidos seleccionando productos de negocios aliados o solicitando servicios personalizados como mandados o encargos.	Cliente
RF-05	Asignación de Repartidor	El sistema permite que los repartidores disponibles visualicen y acepten pedidos pendientes, quedando asignados al pedido correspondiente.	Sistema / Repartidor
RF-06	Actualización de Estado de Pedido	El repartidor puede actualizar el estado del pedido durante el proceso de entrega (pendiente, preparando, en ruta y entregado).	Repartidor
RF-07	Métodos de Pago	El sistema permite al cliente registrar el pago del pedido mediante el método de pago en efectivo.	Cliente

ID del Requisito	Nombre del Requisito	Descripción del Requisito	Usuario
RF-08	Seguimiento del Pedido	El sistema permite al cliente visualizar el estado actual de su pedido desde la aplicación en tiempo real.	Cliente
RF-09	Visualización de Pedidos Disponibles	El sistema muestra a los repartidores los pedidos pendientes que aún no han sido aceptados.	Repartidor
RF-10	Monitoreo de Pedidos	El sistema permite al administrador visualizar los pedidos realizados y su estado dentro de la aplicación.	Administrador
RF-11	Visualización de Pedidos Asignados	El sistema permite al repartidor visualizar los pedidos asignados, incluyendo información del cliente, dirección de entrega y detalles del pedido.	Repartidor

Nota* Elaboración propia basada en los requerimientos funcionales definidos para la aplicación AI-Instante App.

2. Requerimientos no funcionales (RNF)

Los requerimientos no funcionales definen las características de calidad, comportamiento y restricciones técnicas del sistema AI-Instante App. Estos requisitos garantizan que la aplicación no solo cumpla con sus funcionalidades principales, sino que también ofrezca un funcionamiento estable, seguro y una experiencia adecuada para los usuarios.

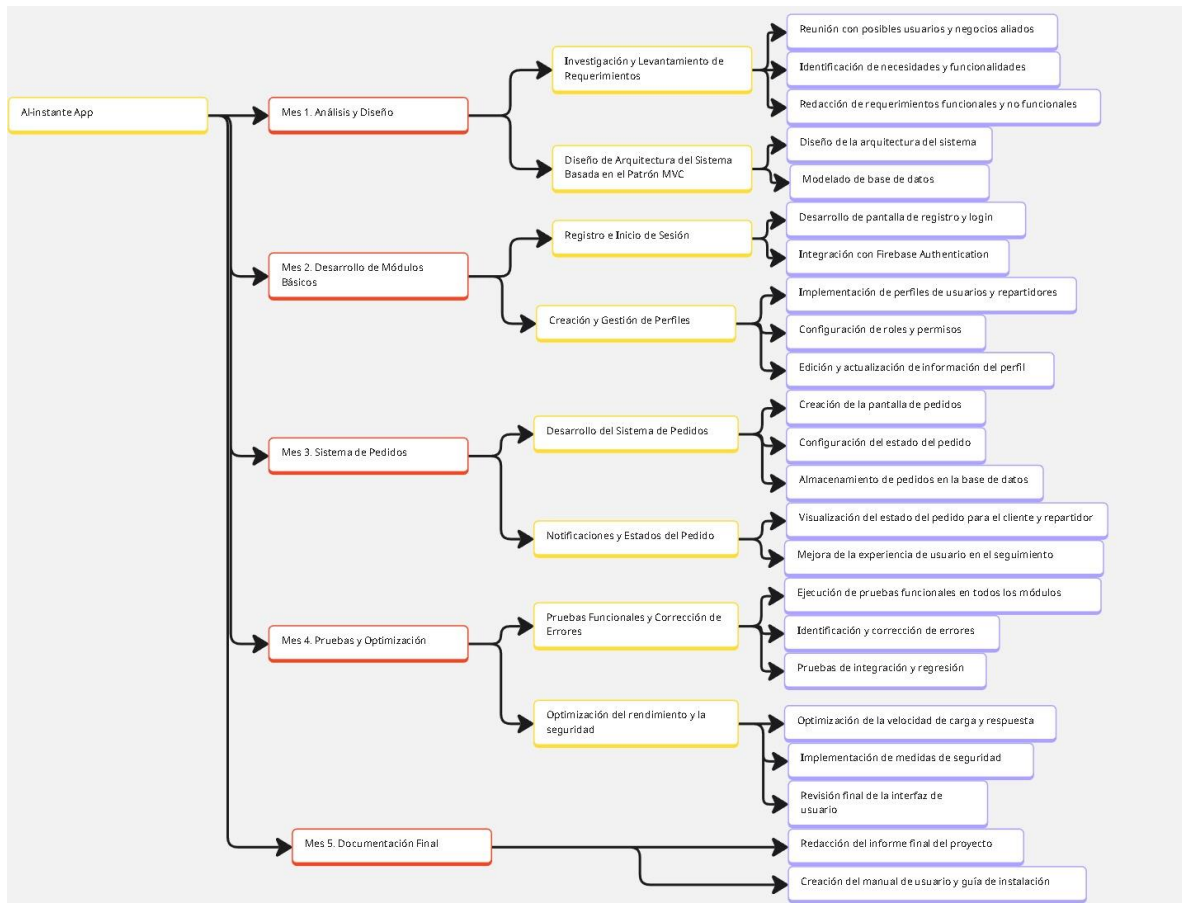
Tabla 4. *Requerimientos no funcionales*

ID del Requisito	Descripción del Requisito
RNF-01	Usabilidad: La aplicación debe contar con una interfaz sencilla, intuitiva y fácil de usar, tanto para clientes como para repartidores, permitiendo una navegación clara entre las funcionalidades.

ID del Requisito	Descripción del Requisito
RNF-02	Rendimiento: La aplicación debe responder en un tiempo adecuado al realizar acciones como enviar pedidos, actualizar estados o cargar listas de información.
RNF-03	Disponibilidad: El sistema debe estar disponible de manera continua para los usuarios, salvo en casos de mantenimiento programado o fallos externos de conectividad.
RNF-04	Escalabilidad: El sistema debe permitir la incorporación de nuevos comercios, productos, usuarios y repartidores sin requerir cambios significativos en la arquitectura existente.
RNF-05	Seguridad: La información personal de los usuarios debe protegerse mediante autenticación segura por número telefónico (OTP) y almacenamiento en bases de datos con control de acceso.
RNF-06	Portabilidad: La aplicación debe poder ejecutarse en dispositivos móviles con sistema operativo Android y iOS.
RNF-07	Mantenibilidad: El código fuente debe estar organizado y estructurado de forma que facilite futuras actualizaciones, mejoras o correcciones del sistema.
RNF-08	Compatibilidad: La aplicación debe funcionar correctamente en versiones recientes de los sistemas operativos móviles compatibles, garantizando estabilidad en distintos dispositivos.

Nota* Elaboración propia basada en los requerimientos no funcionales definidos para la aplicación Al-Instante App.

Figura 8. Planificación del Proyecto



Fuente: Elaboración Propia

Matriz de trazabilidad

Esta matriz permite evidenciar la correspondencia entre cada requisito funcional, su respectivo caso de uso, la evidencia de implementación en la aplicación móvil y el caso de prueba ejecutado para validar su correcto funcionamiento.

Tabla 5. Matriz de trazabilidad de requisitos

Requisito	Caso de Uso	Evidencia en la App	Caso de Prueba
RF-01 Autenticación con OTP	CU-01 Inicio de sesión con verificación OTP	login_screen.dart, otps_creen.dart, Firebase Authentication	CP-01: Ingresar número válido - recibir código OTP - validar acceso según rol. Resultado esperado: acceso correcto según tipo de usuario.
RF-02 Registro de Usuario	CU-02 Registro de cliente	userregistrationscreen.dart, guardado en Firestore	CP-02: Registrar nuevo usuario con datos obligatorios - verificar almacenamiento en colección usuarios.
RF-03 Gestión de Cuentas	CU-03 Gestión de usuarios	Visualización en colección Firestore / módulo administrador	CP-03: Verificar que el administrador puede visualizar datos registrados correctamente.
RF-04 Realizar Pedidos	CU-04 Crear pedido desde carrito	carritoscreen.dart, colección pedidos en Firestore	CP-04: Crear pedido - verificar que se guarda con uid, estado y timestamp.
RF-05 Asignación de Repartidor	CU-05 Aceptar pedido	delivery_order_list.dart, actualización campo repartidorId	CP-05: Repartidor acepta pedido - estado cambia a “preparando” o “en proceso”.
RF-06 Actualización de Estado	CU-06 Actualizar estado del pedido	delivery_assigned_orders.dart	CP-06: Cambiar estado (preparando - en ruta - entregado) - reflejarse en app del cliente.
RF-07 Método de Pago	CU-07 Registrar pago en efectivo	Campo metodoPago en pedido	CP-07: Seleccionar pago en efectivo - verificar que se almacene correctamente.

Requisito	Caso de Uso	Evidencia en la App	Caso de Prueba
RF-08 Seguimiento del Pedido	CU-08 Visualizar estado del pedido	user_orders_screen.dart	CP-08: Cliente consulta pedido - visualizar estado actualizado en tiempo real.
RF-09 Pedidos Disponibles	CU-09 Visualizar pedidos pendientes	delivery_order_list.dart filtrando estado = pendiente	CP-09: Repartidor visualiza pedidos no aceptados.
RF-10 Monitoreo de Pedidos	CU-10 Supervisión de pedidos	Panel administrador / colección pedidos	CP-10: Administrador visualiza pedidos con su estado actual.
RF-11 Pedidos Asignados	CU-11 Ver pedidos asignados	delivery_assigned_orders.dart	CP-11: Repartidor visualiza pedidos asignados con datos del cliente.

Nota* Se presenta la trazabilidad entre requisitos, implementación y validación del sistema desarrollado.

Conclusión de la fase de análisis

A partir de la revisión del contexto y la recolección de información en la comunidad de Palestina, fue posible identificar las principales necesidades y expectativas relacionadas con el uso de una aplicación de domicilios. Este proceso permitió comprender las particularidades del entorno y las dinámicas entre clientes, repartidores y administradores dentro del sistema Al-Instante App.

Como resultado de este análisis, se definieron los distintos tipos de usuarios que interactúan con la plataforma, estableciendo claramente sus roles, funciones y niveles de acceso. De igual forma, se realizó el levantamiento de requerimientos funcionales y no funcionales, los cuales describen tanto las funcionalidades que debe ofrecer el sistema como los criterios de calidad relacionados con usabilidad, seguridad, rendimiento y escalabilidad.

La información obtenida durante esta fase proporcionó una base sólida para orientar las decisiones de diseño y desarrollo de la aplicación, asegurando que la solución propuesta se adapte al contexto real de uso y responda de manera adecuada a las necesidades identificadas en la comunidad, sin perder de vista la posibilidad de evolución futura del sistema.

Fase de Diseño

Durante la fase de diseño se definió la estructura general y el comportamiento interno de la aplicación AI-Instante App, estableciendo las bases técnicas necesarias para su desarrollo. Para ello, se adoptó una arquitectura por capas inspirada en el patrón Modelo–Vista–Controlador (MVC), adaptada al entorno de desarrollo Flutter, lo que permitió separar de manera clara la presentación de la aplicación, la lógica de negocio y el manejo de los datos.

En este contexto, la Vista corresponde a las pantallas e interfaces de usuario desarrolladas en Flutter, el Modelo está representado por la estructura de los datos almacenados en la base de datos en la nube (Firebase Firestore), y la lógica de control se

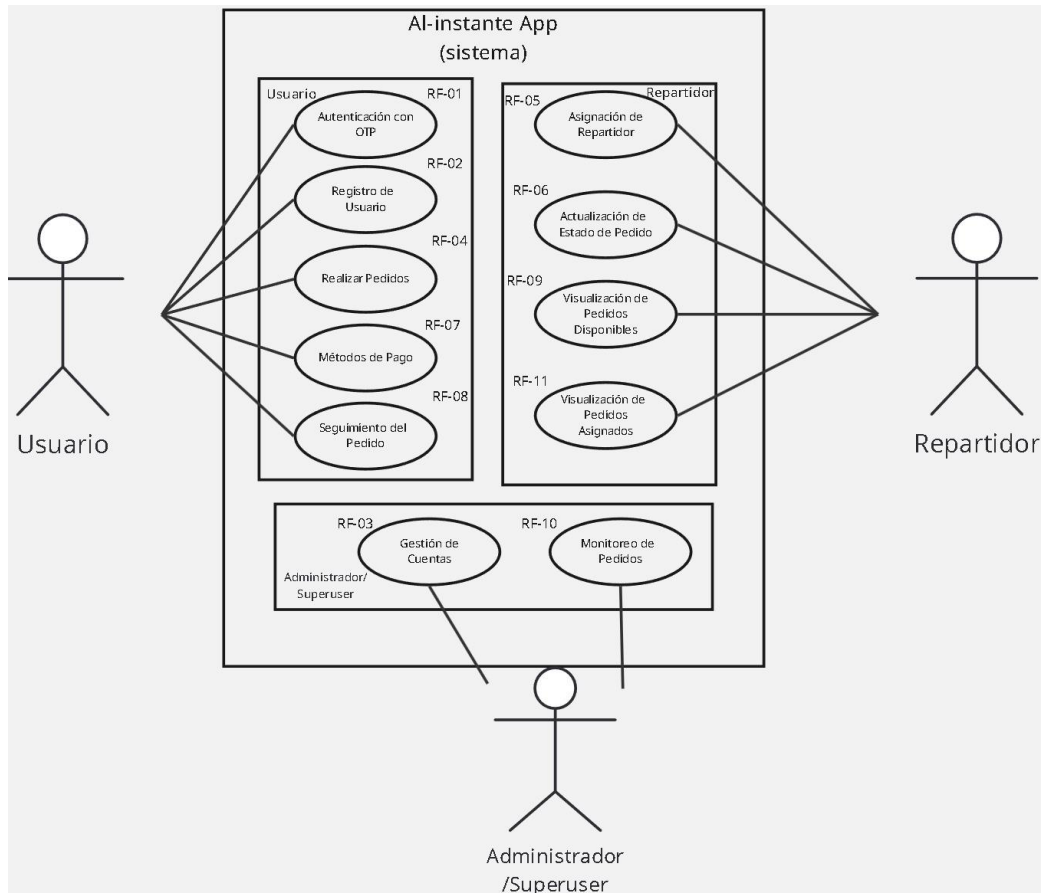
implementa mediante componentes de gestión de estado y servicios que coordinan la interacción entre la vista y los datos.

Como parte del proceso de diseño, se elaboraron diagramas de casos de uso que representan las principales interacciones entre los distintos tipos de usuarios (clientes, repartidores y administradores) y el sistema. Asimismo, se definió un modelo conceptual de datos, que permitió organizar la información relacionada con usuarios, pedidos, estados y direcciones de entrega, sirviendo como base para la implementación de la base de datos en Firestore.

Adicionalmente, se diseñaron los flujos de navegación y las pantallas principales de la aplicación, teniendo en cuenta principios de experiencia de usuario (UX), con el objetivo de facilitar una interacción intuitiva y eficiente. Para la elaboración de los diagramas y modelos visuales se utilizaron herramientas de diseño colaborativo, lo que permitió estructurar y validar el diseño antes de iniciar la fase de desarrollo.

Esta fase fue clave para traducir los requerimientos definidos previamente en un diseño técnico claro y viable, que orientó de manera precisa la implementación de la aplicación.

Figura 9. Diagrama de casos de uso general de la aplicación Al-Instante App



Fuente: Elaboración Propia

Documentación diagrama de casos de uso

El diagrama de casos de uso de Al-Instante App presenta de manera clara y estructurada las principales interacciones entre los actores del sistema: Cliente, Repartidor y Administrador. Este diagrama permite visualizar las funcionalidades disponibles para cada tipo de usuario y la forma en que estas se relacionan con el sistema, facilitando la comprensión del alcance funcional de la aplicación.

A través del diagrama se identifican acciones fundamentales como la autenticación mediante código OTP, el registro de usuarios, la realización y seguimiento de pedidos por parte del cliente, la visualización y actualización del estado de los pedidos por parte del repartidor, así como el monitoreo general y la gestión de usuarios realizada por el administrador. Estas interacciones reflejan de forma directa los requerimientos funcionales definidos en la fase de análisis.

Además, el diagrama de casos de uso sirve como una herramienta de apoyo para la comunicación entre los desarrolladores y las partes interesadas, ya que permite validar que las funcionalidades del sistema estén alineadas con los roles y necesidades de cada actor. De esta manera, contribuye a un diseño ordenado y coherente del sistema, asegurando que cada usuario disponga de las funcionalidades necesarias para desempeñar adecuadamente su rol dentro de la aplicación.

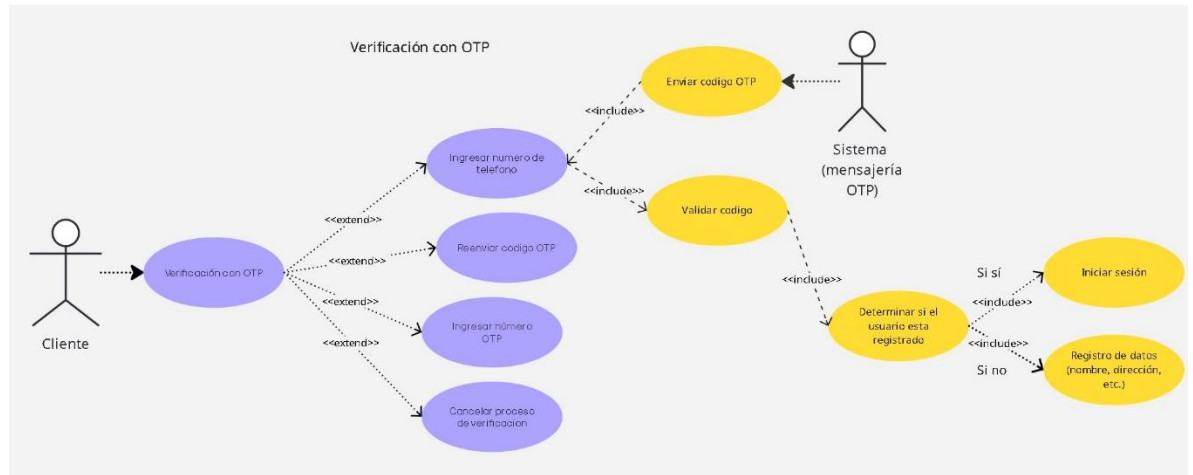
Casos de uso extendidos del sistema

A partir del análisis de los requerimientos funcionales y del diagrama de casos de uso presentado anteriormente, se identificaron los casos de uso principales de la aplicación Al-Instante App. Estos casos de uso representan los procesos más relevantes y críticos del sistema, ya que describen los flujos esenciales que permiten la interacción entre los distintos actores y la plataforma.

Debido a la complejidad y alcance del sistema, no todos los requerimientos funcionales fueron documentados como casos de uso extendidos. En su lugar, se seleccionaron aquellos que concentran la mayor parte de la lógica del sistema y que permiten comprender de forma clara el funcionamiento general de la aplicación, incluyendo la autenticación, el registro de usuarios, la realización y gestión de pedidos, la asignación de repartidores y el seguimiento del estado de las entregas.

La documentación de los casos de uso extendidos se presenta a continuación, detallando para cada uno su actor principal, condiciones, flujos de ejecución y reglas de negocio, con el fin de ofrecer una visión precisa del comportamiento del sistema ante diferentes escenarios.

Figura 10. Diagrama de caso de uso extendido (verificación OTP).



Fuente: Elaboración Propia

Tabla 6. Verificación OTP

Elemento	Descripción
ID	CU-01
Nombre	Verificación de número por OTP
Actor principal	Cliente / Repartidor / Administrador
Actores secundarios	Servicio de autenticación (Firebase Authentication – OTP)
Breve descripción	El actor ingresa su número de teléfono, recibe un código OTP por SMS, lo introduce en la aplicación y el sistema valida su autenticidad. Tras la validación, el sistema identifica el rol del usuario (cliente, repartidor o administrador) y lo redirige a la pantalla correspondiente o al flujo de registro si aplica.
Precondiciones	<ol style="list-style-type: none"> 1. El actor dispone de un número de teléfono válido y activo. 2. La aplicación cuenta con conexión a internet.

Elemento	Descripción
Postcondiciones	<p>3. El servicio de envío de OTP se encuentra disponible.</p> <p>1. El número telefónico queda verificado mediante OTP y el usuario queda autenticado en la aplicación.</p> <p>2. El sistema determina el rol y redirige al módulo correspondiente.</p> <p>3. Si el usuario cliente no tiene perfil registrado, se inicia el flujo de registro de datos.</p>
Flujo principal	<p>1. El actor selecciona la opción Iniciar sesión.</p> <p>2. El sistema solicita el número telefónico.</p> <p>3. El actor ingresa el número en el formato requerido (ej. +57...).</p> <p>4. El sistema envía el código OTP vía SMS.</p> <p>5. El actor recibe el SMS con el código OTP.</p> <p>6. El actor ingresa el código OTP en la aplicación.</p> <p>7. El sistema valida el OTP y autentica al usuario.</p> <p>8. El sistema consulta los registros y determina el rol (administrador, repartidor o cliente).</p> <p>9. El sistema redirige al actor a la pantalla correspondiente según el rol.</p> <p>10. Si el actor es cliente y no tiene perfil completo, el sistema redirige al registro de usuario.</p>
Flujos alternativos	<p>A1. Código incorrecto: el sistema muestra un mensaje de error y permite reintentar la verificación.</p> <p>A2. No llega el código / expira: el sistema permite reiniciar el proceso de verificación solicitando nuevamente el OTP.</p> <p>A3. Cancelación del proceso: el actor abandona la verificación y el sistema retorna a la pantalla anterior sin autenticar.</p>
Reglas de negocio	<p>A4. Fallo de conectividad: el sistema notifica la falta de conexión y bloquea temporalmente la verificación hasta recuperar internet.</p> <ul style="list-style-type: none"> - El número de teléfono debe registrarse en formato internacional (ej. +57 para Colombia). - El OTP es temporal y su validez está determinada por el servicio de autenticación. - El sistema debe limitar reintentos consecutivos para evitar abuso del proceso de verificación.

Nota* Elaboración propia basada en el requerimiento funcional RF-01 y en el flujo de autenticación implementado en la aplicación AI-Instante App.

Figura 11. Diagrama de caso de uso extendido (Registro de usuario).



Fuente: Elaboración Propia

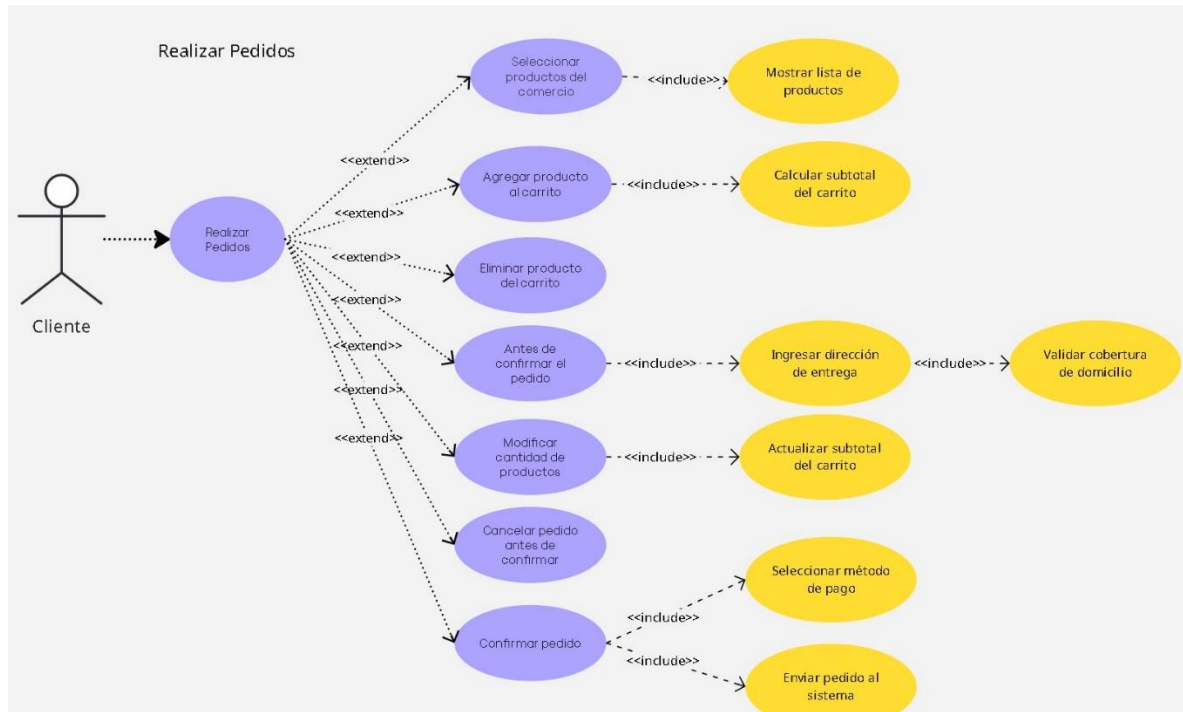
Tabla 7. Registro de usuario

Elemento	Descripción
ID	CU-02
Nombre	Registro de usuario
Actor principal	Cliente
Actores secundarios	Sistema
Breve descripción	El sistema permite al cliente registrar su información personal básica después de haber validado su número telefónico mediante OTP, con el fin de completar su perfil y habilitar la realización de pedidos.
Precondiciones	<ol style="list-style-type: none"> 1. El cliente ha verificado exitosamente su número telefónico mediante OTP. 2. El cliente no cuenta con un perfil previamente registrado en el sistema. 3. La aplicación dispone de conexión a internet.
Postcondiciones	<ol style="list-style-type: none"> 1. El perfil del cliente queda registrado en la base de datos del sistema. 2. La información de contacto y dirección queda asociada al número telefónico validado. 3. El cliente queda habilitado para acceder a las funcionalidades de realización de pedidos.

Elemento	Descripción
Flujo principal	<ol style="list-style-type: none"> 1. El sistema detecta que el cliente no tiene un perfil registrado. 2. El sistema muestra el formulario de registro. 3. El cliente ingresa su nombre. 4. El cliente ingresa el barrio de residencia. 5. El cliente ingresa la dirección de entrega. 6. El cliente ingresa una descripción opcional como referencia de ubicación. 7. El cliente confirma el registro. 8. El sistema valida la información ingresada. 9. El sistema almacena los datos del cliente en la base de datos. 10. El sistema redirige al cliente a la pantalla principal de la aplicación.
Flujos alternativos	<p>A1. Datos incompletos: el sistema muestra un mensaje indicando los campos obligatorios pendientes y no permite continuar.</p> <p>A2. Cancelación del registro: el cliente cancela el proceso y el sistema retorna a la pantalla anterior sin guardar la información.</p>
Reglas de negocio	<p>A3. Error de conexión: el sistema notifica la falla y bloquea el registro hasta que se restablezca la conexión.</p> <ul style="list-style-type: none"> - El registro de usuario solo aplica para clientes. - El número telefónico no puede modificarse durante el registro, ya que fue validado previamente. - Los campos nombre, barrio y dirección son obligatorios. - La descripción es opcional y se utiliza como referencia para la entrega.

Nota* Elaboración propia basada en el requerimiento funcional RF-02 y en el proceso de registro implementado en la aplicación AI-Instante App.

Figura 12. Diagrama de caso de uso extendido (Realizar pedido).



Fuente: Elaboración Propia

Tabla 8. Realizar pedido

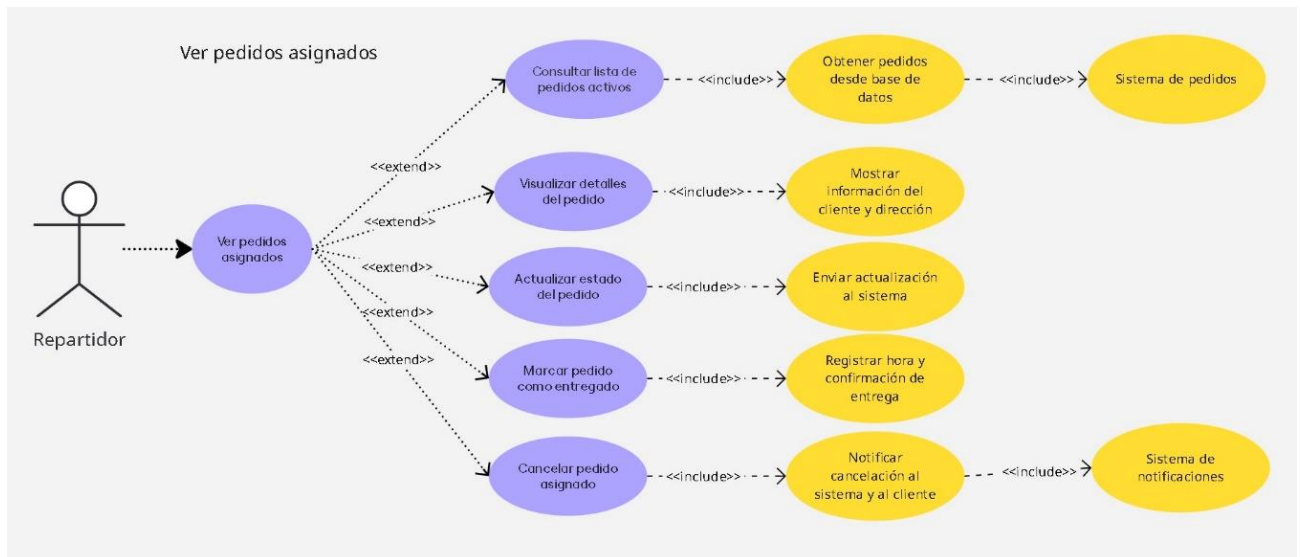
Elemento	Descripción
ID	CU-03
Nombre	Realizar pedido
Actor principal	Cliente
Actores secundarios	Sistema
Breve descripción	El cliente selecciona productos de comercios aliados o registra solicitudes personalizadas (mandado o encargo), confirma el pedido, valida su dirección de entrega y el sistema guarda el pedido con su información, costos y estado inicial para que pueda ser atendido por un repartidor.
Precondiciones	1. El cliente está autenticado y cuenta con perfil registrado.

Elemento	Descripción
Postcondiciones	<p>2. La aplicación tiene conexión a internet.</p> <p>3. El cliente tiene al menos un producto en el carrito o una solicitud personalizada registrada (mandado/encargo).</p> <p>4. El dispositivo tiene permisos de ubicación habilitados para calcular zona y costo de domicilio.</p> <p>1. El pedido queda registrado en la base de datos del sistema con estado inicial “pendiente de repartidor”.</p> <p>2. Queda almacenada la información de entrega (dirección, barrio, referencia y coordenadas).</p> <p>3. Se registra el detalle del pedido (productos/solicitud personalizada, subtotal, domicilio, recargos y total).</p> <p>4. El carrito queda limpio tras confirmar el pedido.</p>
Flujo principal	<p>1. El cliente explora un comercio aliado o servicio y selecciona productos, agregándolos al carrito.</p> <p>2. (Opcional) El cliente registra una solicitud personalizada como mandado o encargo personalizado.</p> <p>3. El cliente ingresa al módulo Carrito y selecciona Confirmar pedido.</p> <p>4. El sistema valida que exista contenido en el pedido (productos o solicitud personalizada).</p> <p>5. El sistema obtiene la ubicación actual del cliente (latitud/longitud) y calcula la zona y el costo de domicilio.</p> <p>6. El sistema muestra un resumen del pedido (productos, costos y total) y permite seleccionar el método de pago (efectivo).</p> <p>7. El cliente confirma continuar.</p> <p>8. El sistema muestra un formulario para confirmar o editar la dirección de entrega (barrio, dirección y referencia).</p> <p>9. El cliente confirma la dirección.</p> <p>10. El sistema registra el pedido en la base de datos con la información del cliente, entrega, productos/solicitudes, costos y estado “pendiente de repartidor”.</p> <p>11. El sistema notifica al cliente que el pedido fue enviado correctamente y limpia el carrito.</p>
Flujos alternativos	<p>A1. Carrito vacío: el sistema informa que no hay elementos para enviar y cancela el proceso.</p> <p>A2. GPS desactivado / sin permisos: el sistema informa que no pudo obtener ubicación y detiene el proceso hasta que se habilite.</p> <p>A3. Fuera de cobertura: el sistema informa que la ubicación está fuera del área de servicio y cancela el envío del pedido.</p>

Elemento	Descripción
Reglas de negocio	A4. Cancelación por el cliente: el cliente cancela en cualquiera de los diálogos de confirmación y el sistema no registra el pedido.
	A5. Error al guardar: el sistema informa el error y mantiene el carrito sin cambios para reintentar.
	<ul style="list-style-type: none"> - El pedido debe contener al menos un producto o una solicitud personalizada (mandado/encargo). - El domicilio se calcula con base en la ubicación actual del cliente y la zona definida por el sistema. - El método de pago permitido es efectivo. - El estado inicial del pedido es “pendiente de repartidor”. - La dirección de entrega debe confirmarse antes de enviar el pedido.

Nota* Elaboración propia basada en el requerimiento funcional RF-04 y en el flujo de creación y confirmación de pedidos implementado en la aplicación AI-Instante App.

Figura 13. Diagrama de caso de uso extendido (Asignación de repartidor).



Fuente: Elaboración Propia

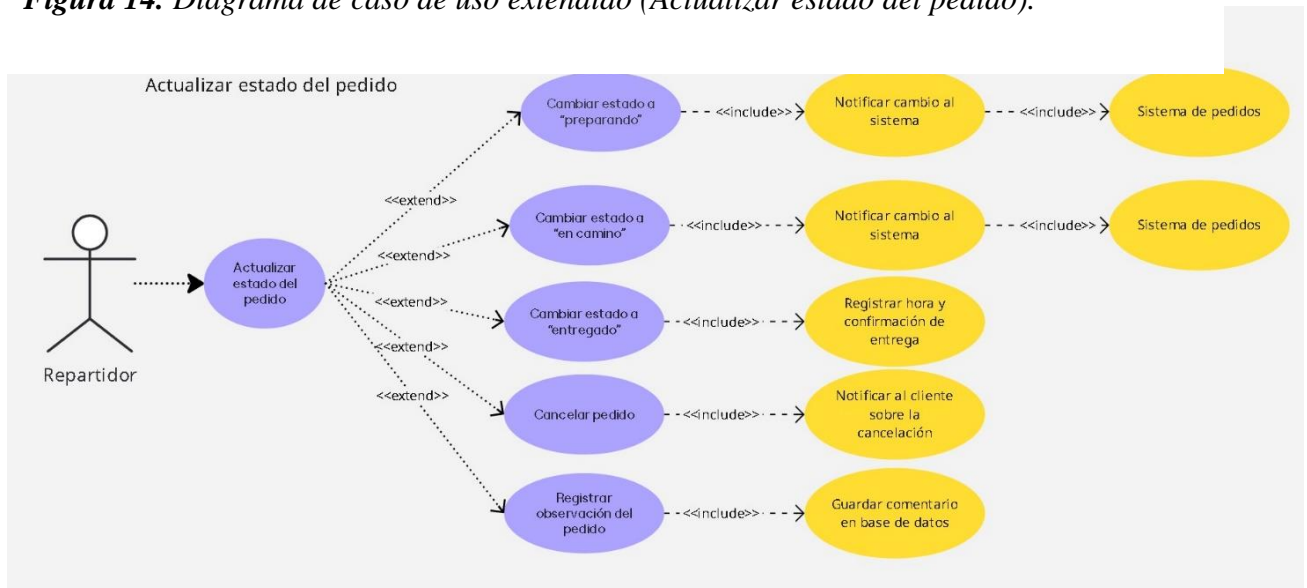
Tabla 9. Asignación de repartidor

Elemento	Descripción
ID	CU-04
Nombre	Asignación de repartidor (aceptación de pedido)
Actor principal	Repartidor
Actores secundarios	Sistema
Breve descripción	El repartidor visualiza la lista de pedidos pendientes y selecciona uno para aceptarlo. El sistema asigna el pedido al repartidor autenticado y actualiza el estado para iniciar el proceso de entrega.
Precondiciones	<ol style="list-style-type: none"> 1. El repartidor está autenticado en la aplicación. 2. El repartidor se encuentra registrado en el sistema y con acceso al módulo de repartidor. 3. Existen pedidos con estado “pendiente de repartidor”. 4. La aplicación tiene conexión a internet.
Postcondiciones	<ol style="list-style-type: none"> 1. El pedido queda asignado al repartidor autenticado (se registra el identificador del repartidor). 2. El estado del pedido cambia de “pendiente de repartidor” a “preparando” (pedido aceptado). 3. Queda registrada la hora de asignación del pedido. 4. El pedido aparece en la sección de “Mis pedidos” del repartidor.
Flujo principal	<ol style="list-style-type: none"> 1. El repartidor ingresa al módulo Pedidos. 2. El sistema muestra la lista de pedidos disponibles con estado “pendiente de repartidor”. 3. El repartidor selecciona un pedido para ver su detalle. 4. El sistema muestra la información del pedido (cliente, dirección, referencia, productos/solicitud personalizada y total). 5. El repartidor selecciona la opción Aceptar pedido. 6. El sistema registra el identificador del repartidor en el pedido y actualiza el estado a “preparando”. 7. El sistema confirma la asignación y el pedido queda disponible en Mis pedidos. <p>A1. Pedido ya no disponible: si otro repartidor aceptó el pedido antes, el sistema informa que ya fue asignado y retorna a la lista.</p>
Flujos alternativos	<p>A2. Error de conexión: el sistema notifica la falla y evita completar la asignación hasta restablecer internet.</p> <p>A3. Error al actualizar pedido: el sistema informa el error y no modifica el estado ni la asignación.</p>

Elemento	Descripción
Reglas de negocio	<p>A4. Cancelación: el repartidor sale del detalle sin aceptar y el pedido permanece pendiente de repartidor.</p> <ul style="list-style-type: none"> - Solo los pedidos en estado “pendiente de repartidor” pueden ser aceptados. - Al aceptar el pedido, se registra el repartidor asignado y la hora de asignación. - Un pedido solo puede estar asignado a un repartidor a la vez.

Nota: Elaboración propia basada en el requerimiento funcional RF-05 y en el flujo de aceptación y asignación de pedidos implementado en la aplicación AI-Instante App.

Figura 14. Diagrama de caso de uso extendido (Actualizar estado del pedido).



Fuente: Elaboración Propia

Tabla 10. Actualización del estado del pedido

Elemento	Descripción
ID	CU-05
Nombre	Actualización del estado del pedido
Actor principal	Repartidor
Actores secundarios	Sistema
Breve descripción	El repartidor actualiza el estado del pedido durante el proceso de entrega, permitiendo que el sistema y el cliente conozcan en tiempo real el avance del pedido.
Precondiciones	<ol style="list-style-type: none"> 1. El repartidor está autenticado en la aplicación. 2. El pedido ya fue aceptado y está asignado al repartidor. 3. El pedido se encuentra en un estado válido para actualización (preparando, en ruta). 4. La aplicación cuenta con conexión a internet.
Postcondiciones	<ol style="list-style-type: none"> 1. El estado del pedido se actualiza correctamente en la base de datos. 2. El nuevo estado queda visible para el cliente en la aplicación. 3. El pedido avanza en el flujo hasta llegar al estado “entregado”.
Flujo principal	<ol style="list-style-type: none"> 1. El repartidor accede a la sección Mis pedidos. 2. El sistema muestra la lista de pedidos asignados al repartidor. 3. El repartidor selecciona un pedido para ver su detalle. 4. El sistema muestra el estado actual del pedido. 5. El repartidor selecciona la opción para cambiar el estado del pedido. 6. El sistema actualiza el estado del pedido según la acción realizada: preparando – en ruta- entregado. 7. El sistema guarda la actualización y notifica el cambio al cliente.
Flujos alternativos	<p>A1. Estado no válido: si el repartidor intenta cambiar a un estado no permitido, el sistema bloquea la acción.</p> <p>A2. Error de conexión: el sistema muestra un mensaje de error y no registra el cambio.</p> <p>A3. Pedido ya finalizado: si el pedido está en estado entregado, el sistema no permite más modificaciones.</p>
Reglas de negocio	<ul style="list-style-type: none"> - Los estados del pedido deben seguir un orden secuencial. - Solo el repartidor asignado puede actualizar el estado del pedido. - Una vez marcado como entregado, el pedido no puede ser modificado.

Elemento	Descripción
Actores secundarios	Sistema
Breve descripción	El cliente puede consultar en tiempo real el estado de su pedido desde la aplicación, visualizando el avance del proceso de entrega y la información básica del pedido.
Precondiciones	<ol style="list-style-type: none"> 1. El cliente está autenticado en la aplicación. 2. El cliente ha realizado al menos un pedido. 3. El pedido se encuentra registrado en el sistema. 4. La aplicación tiene conexión a internet.
Postcondiciones	<ol style="list-style-type: none"> 1. El cliente visualiza el estado actualizado del pedido. 2. El cliente puede acceder al detalle del pedido en cualquier momento. 3. El seguimiento finaliza cuando el pedido alcanza el estado “entregado”.
Flujo principal	<ol style="list-style-type: none"> 1. El cliente accede a la opción Mis pedidos desde la aplicación. 2. El sistema muestra la lista de pedidos asociados al usuario. 3. El cliente selecciona un pedido específico. 4. El sistema muestra el estado actual del pedido (pendiente, preparando, en ruta o entregado). 5. El sistema actualiza automáticamente el estado del pedido conforme el repartidor realiza cambios. <p>A1. Sin pedidos registrados: el sistema muestra un mensaje indicando que no existen pedidos asociados al usuario.</p>
Flujos alternativos	<p>A2. Error de conexión: el sistema informa al usuario que no es posible cargar la información en ese momento.</p>
Reglas de negocio	<p>A3. Pedido cancelado: el sistema muestra el estado cancelado y bloquea el seguimiento del flujo normal.</p> <ul style="list-style-type: none"> - El cliente solo puede visualizar pedidos asociados a su cuenta. - El estado mostrado al cliente depende exclusivamente de las actualizaciones realizadas por el repartidor. - El seguimiento se mantiene disponible incluso después de que el pedido haya sido entregado, como parte del historial.

Nota* Elaboración propia basada en el requerimiento funcional RF-08 y en el módulo de visualización de estados de pedidos de la aplicación AI-Instante App.

Arquitectura de la aplicación basada en el patrón MVC adaptado

La aplicación AI-Instante App fue diseñada siguiendo una arquitectura por capas inspirada en el patrón Modelo–Vista–Controlador (MVC), adaptada al entorno de desarrollo Flutter. Aunque Flutter no implementa de manera estricta el patrón MVC tradicional, sí permite una separación clara de responsabilidades entre la interfaz de usuario, la gestión de datos y la lógica de control, lo cual facilita el análisis, mantenimiento y escalabilidad del sistema.

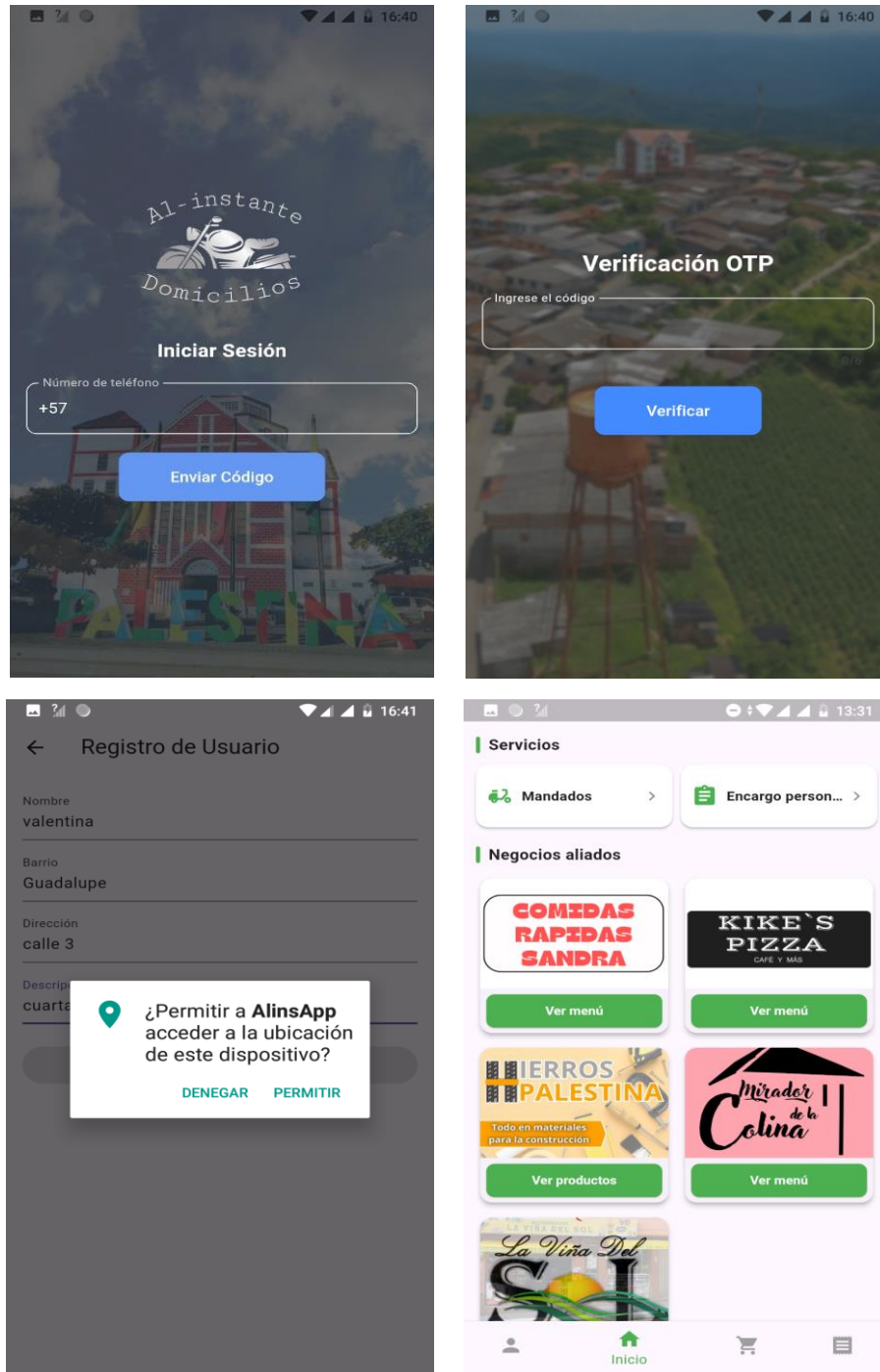
A continuación, se describen los componentes principales de esta arquitectura y su aplicación dentro del desarrollo de la plataforma.

Vista (View)

La vista corresponde a la capa de presentación de la aplicación, es decir, a todas las pantallas y componentes visuales con los que el usuario interactúa directamente. Esta capa se encarga de mostrar la información proveniente del sistema y de capturar las acciones del usuario, como el ingreso de datos, la selección de opciones o la navegación entre pantallas.

En AI-Instante App, la vista está implementada mediante pantallas desarrolladas en Flutter, utilizando widgets que permiten construir interfaces intuitivas y responsivas para cada tipo de usuario (cliente, repartidor y administrador).

Figura 16. Inicio de Sesión, verificación OTP, registro de usuario y menú principal de negocios aliados.



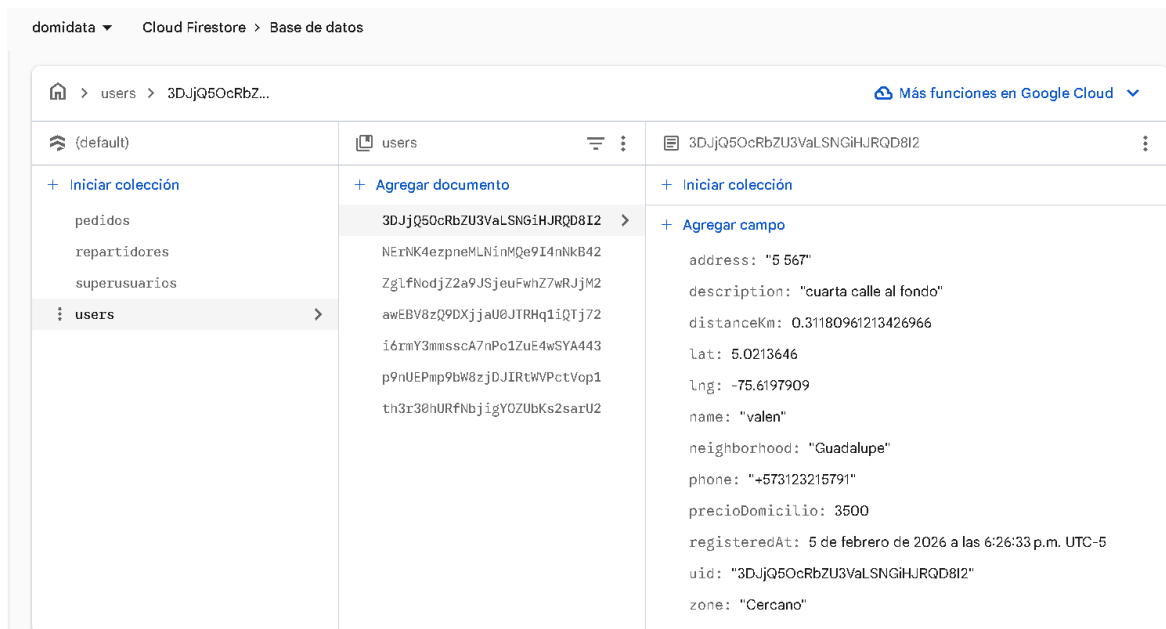
Fuente: Elaboración Propia

Modelo (Model)

El modelo representa la estructura de los datos del sistema y la forma en que estos son almacenados y gestionados. Esta capa contiene la información relacionada con los usuarios, pedidos, productos, direcciones de entrega y estados del pedido, y es fundamental para el correcto funcionamiento de la aplicación.



En el caso de AI-Instante App, el modelo está implementado a través de una base de datos en la nube utilizando Firebase Firestore, donde la información se organiza en colecciones y documentos. Esta estructura permite un acceso eficiente a los datos y una sincronización en tiempo real entre los distintos usuarios del sistema.

Figura 17. Almacenamiento y control de los datos



Fuente: Elaboración Propia

Figura 18. Ejemplo de colección usuarios y colección pedido

3DJjQ5OcRbZU3VaLSNGiHJRQD8I2	2hxyvLYf3jqBSbOCMbYJ
+ Iniciar colección	+ Iniciar colección
+ Agregar campo	+ Agregar campo
address: "5 567"	cantidad: 2
description: "cuarta calle al fondo"	nombre: "Hamburguesa sencilla"
distanceKm: 0.31180961213426966	precio: 14000
lat: 5.0213646	restaurante: "Sandra Comidas"
lng: -75.6197909	recargoExtra: 0
name: "valen"	repartidorId: "BtDNu5lOYwOpjAp7MBI7P4Kug7w2"
neighborhood: "Guadalupe"	subtotal: 28000
phone: "+573123215791"	telefono: "+573123215765"
precioDomicilio: 3500	telefonoRepartidor: "+573008523744"
registeredAt: 5 de febrero de 2026 a las 6:26:33 p.m. UTC-5	timestamp: 13 de enero de 2026 a las 9:14:59 p.m. UTC-5
uid: "3DJjQ5OcRbZU3VaLSNGiHJRQD8I2"	total: 31500 (number)  
zone: "Cercano"	uid: "th3r30hURfNbjigYOZUbKs2sarU2"

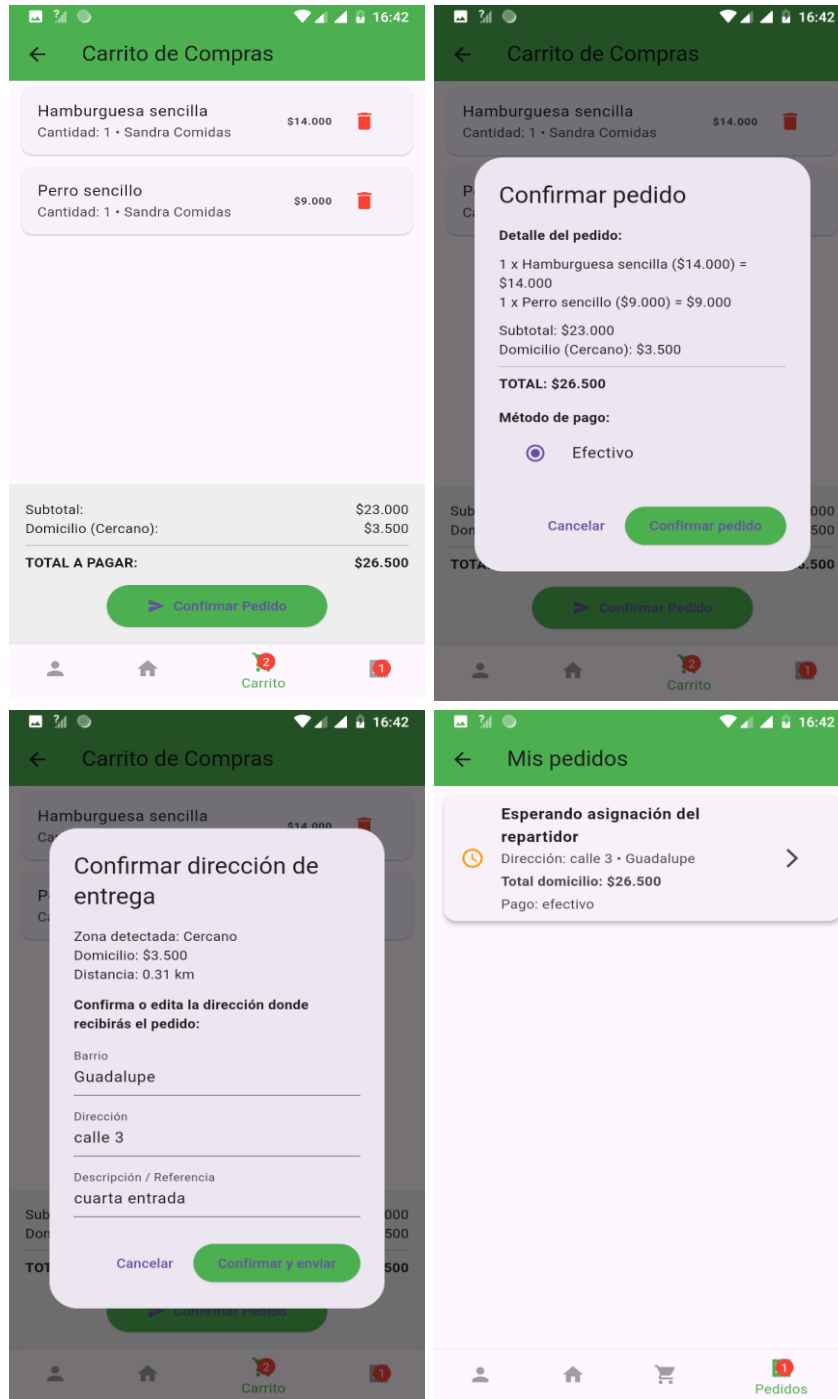
Fuente: Elaboración Propia

Controlador / Lógica de control (Controller)

La lógica de control es la capa encargada de procesar las acciones realizadas por el usuario, coordinar la comunicación entre la vista y el modelo, y gestionar el estado de la aplicación. Esta capa toma decisiones, valida información y ejecuta los procesos necesarios para que el sistema funcione correctamente.

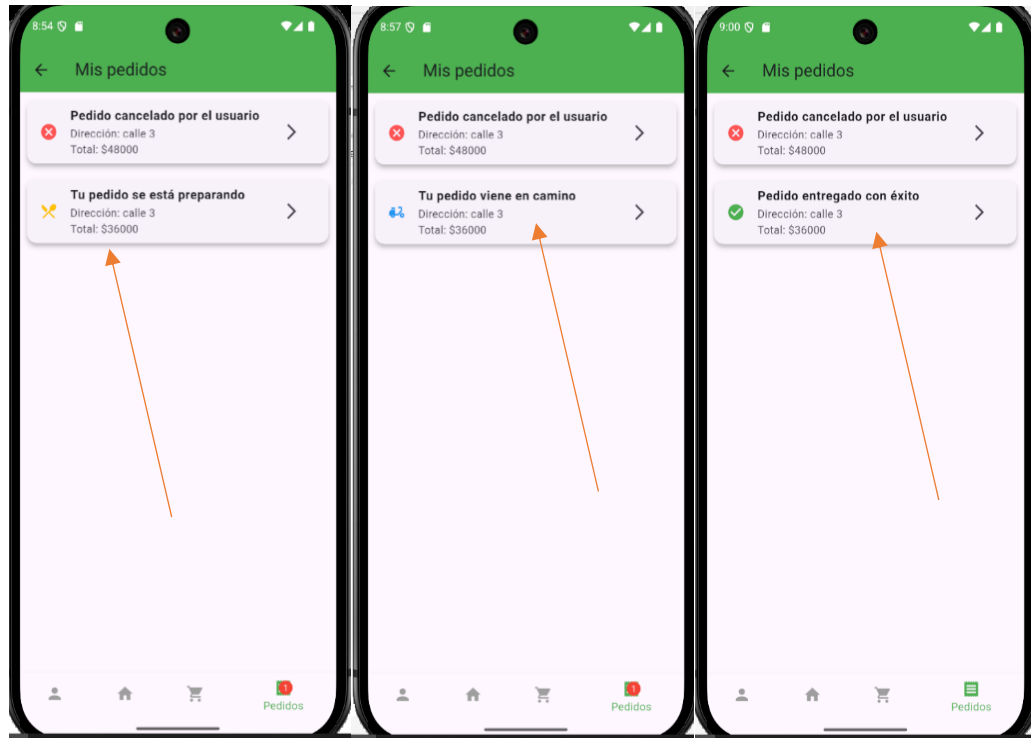
En Al-Instante App, esta función es asumida por componentes de gestión de estado y servicios, los cuales controlan acciones como la autenticación de usuarios, la gestión del carrito de compras, el cálculo del costo de domicilio según la ubicación, la creación de pedidos y la actualización de su estado durante la entrega.

Figura 19. Carrito de Compras, productos seleccionados, confirmación de pedido, confirmación de dirección y esperando asignación de repartidor.



Fuente: Elaboración Propia

Figura 20. Seguimiento del estado del pedido en el usuario, actualizaciones en tiempo real.



Fuente: Elaboración Propia

Conclusión de la Arquitectura

La implementación de una arquitectura basada en MVC adaptado permitió organizar el desarrollo de AI-Instante App de manera estructurada, separando claramente la interfaz de usuario, la gestión de datos y la lógica del sistema. Esta separación facilitó el mantenimiento del código, mejoró la claridad del diseño y permitió una mejor adaptación a los requerimientos definidos, garantizando un sistema escalable y alineado con las necesidades reales del entorno de aplicación.

Diseño de la Base de Datos

El diseño de la base de datos de AI-instante App se implementa utilizando Firebase Cloud Firestore, una base de datos No Relacional (NoSQL) orientada a documentos. Este enfoque permite estructurar la información mediante colecciones y documentos, en lugar de tablas y registros tradicionales como ocurre en los sistemas relacionales. La elección de Firestore responde a la necesidad de contar con una solución flexible, escalable y con sincronización en tiempo real, adecuada para una aplicación móvil de domicilios donde se gestionan múltiples usuarios, pedidos dinámicos y asignaciones de repartidores en constante actualización.

En la figura se presenta un modelo visual de la estructura real implementada en Firestore. Cada bloque representa una colección principal del sistema: users (clientes), repartidores, pedidos y superusuarios. En este modelo, la colección pedidos actúa como eje central del sistema, ya que almacena toda la información relacionada con cada orden, incluyendo datos del cliente, ubicación de entrega, costos, método de pago, estado y fechas de registro.

Aunque Firestore no utiliza relaciones formales ni operaciones tipo *JOIN* como las bases de datos relacionales, se implementan relaciones lógicas mediante el uso de identificadores únicos. Por ejemplo, cada pedido almacena el campo *uid*, el cual corresponde al identificador del cliente en la colección *users*, y opcionalmente almacena el campo *repartidorId*, que referencia al repartidor asignado en la colección *repartidores*. Esta relación permite consultar de forma eficiente el historial de pedidos de un usuario, así como los pedidos asignados a un repartidor, manteniendo una estructura simple y optimizada para el rendimiento.

Adicionalmente, el detalle de los productos del pedido se almacena dentro del mismo documento de pedido mediante el campo productos[], el cual corresponde a un arreglo de mapas que incluye información como nombre, cantidad, precio y restaurante. Esta decisión permite que cada pedido conserve una “fotografía” del contenido exacto solicitado por el cliente en el momento de la compra, evitando inconsistencias futuras en caso de que los productos o precios cambien.

Esta configuración ha permitido gestionar operaciones clave como la autenticación por número telefónico, el registro de usuarios, la asignación dinámica de pedidos a repartidores, y el seguimiento del estado de cada orden en tiempo real, sin necesidad de servidores dedicados. Firebase, al integrar servicios como Authentication, Firestore y funcionalidades complementarias para notificaciones, proporciona una plataforma unificada que simplifica el backend y garantiza un funcionamiento estable en dispositivos móviles.

Tabla 12. *Tabla PK/FK – Colección users (Clientes)*

Campo	Tipo	Clave Referencia / Relación	Descripción
Uid	string	PK Auth UID - users/{uid}	Identificador del usuario
Name	string		Nombre
Telefono	string		Teléfono
neighborhood	string		Barrio
Address	string		Dirección

Campo	Tipo	Clave Referencia / Relación	Descripción
description	string		Referencia/indicaciones
Lat	number		Latitud (ubicación guardada)
Lng	number		Longitud (ubicación guardada)
Zone	string		Zona (ej: Cercano)
precioDomicilio	number		Tarifa sugerida/guardada
registeredAt	timestamp		Fecha de registro

Nota* Esta colección almacena la información principal de los clientes registrados, incluyendo datos de contacto y ubicación. Su identificador principal es el uid generado por Firebase Authentication.

Tabla 13. *Tabla PK/FK – Colección repartidores*

Campo	Tipo	Clave Referencia / Relación	Descripción
Uid	string	PK Auth UID - repartidores/{uid}	Identificador del repartidor
Nombre	string		Nombre completo
Telefono	string		Telefono
Zona	string		Zona donde opera
disponible	boolean		Disponible para pedidos

Nota* Esta colección guarda los datos de los repartidores habilitados en el sistema, incluyendo su zona de operación y disponibilidad. Su identificador principal es el uid del repartidor autenticado.

Tabla 14. *Tabla PK/FK – Colección superusuarios*

Campo	Tipo	Clave	Referencia / Relación	Descripción
uid	string	PK	Auth UID - superusuarios/{uid}	Identificador del superusuario
nombre	string			Nombre
telefono	string			Teléfono
activo	boolean			Estado de habilitación

Nota* Esta colección representa el rol administrador del sistema. Los superusuarios no están ligados directamente a los pedidos, pero tienen permisos para gestionar usuarios, repartidores y pedidos.

Tabla 15. *Tabla PK/FK – Colección pedidos (Documento principal)*

Campo	Tipo	Clave	Referencia / Relación	Descripción
Id	string	PK	Autogenerado - pedidos/{pedidoId}	Identificador del pedido
Uid	string	FK lógica	- users/{uid}	Cliente dueño del pedido
repartidorId	string / null	FK lógica	- repartidores/{uid}	Repartidor asignado (puede ser null)
Cliente	string			Nombre del cliente (snapshot)
Teléfono	string			Teléfono del cliente (snapshot)
telefonoRepartidor	string			Teléfono del repartidor (snapshot, si aplica)
deliveryNeighborhood	string			Barrio de entrega
deliveryAddress	string			Dirección de entrega
deliveryDescription	string			Referencia de entrega
deliveryLat	number			Latitud entrega (GPS)
deliveryLng	number			Longitud entrega (GPS)
distanceKm	number			Distancia del pedido

Campo	Tipo	Clave	Referencia / Relación	Descripción
Zone	string			Zona calculada
precioDomicilio	number			Valor domicilio
Subtotal	number			Suma de productos
recargoExtra	number			Recargo por puntos adicionales
Total	number			Total final
pedidoPersonalizado	string			Mandado / encargo (texto)
metodoPago	string			Ej: efectivo Ej: pendiente de
Estado	string			repartidor / preparando / en ruta / entregado
horaAsignacion	timestamp			Momento de asignación (si lo usas)
timestamp	timestamp			Fecha creación (serverTimestamp)
productos	array(map)			Lista de productos del pedido

Nota* Esta colección almacena cada pedido realizado en la aplicación y funciona como el eje central del sistema. Incluye información del cliente, entrega, costos, estado del pedido y la referencia lógica al repartidor asignado mediante repartidorId (opcional).

Tabla 16. Subtabla: pedidos. productos [] (Array de mapas)

Campo dentro de productos[]	Tipo	Descripción
Nombre	string	Nombre del producto
Cantidad	number	Cantidad comprada
Precio	number	Precio unitario
Restaurante	string	Negocio asociado

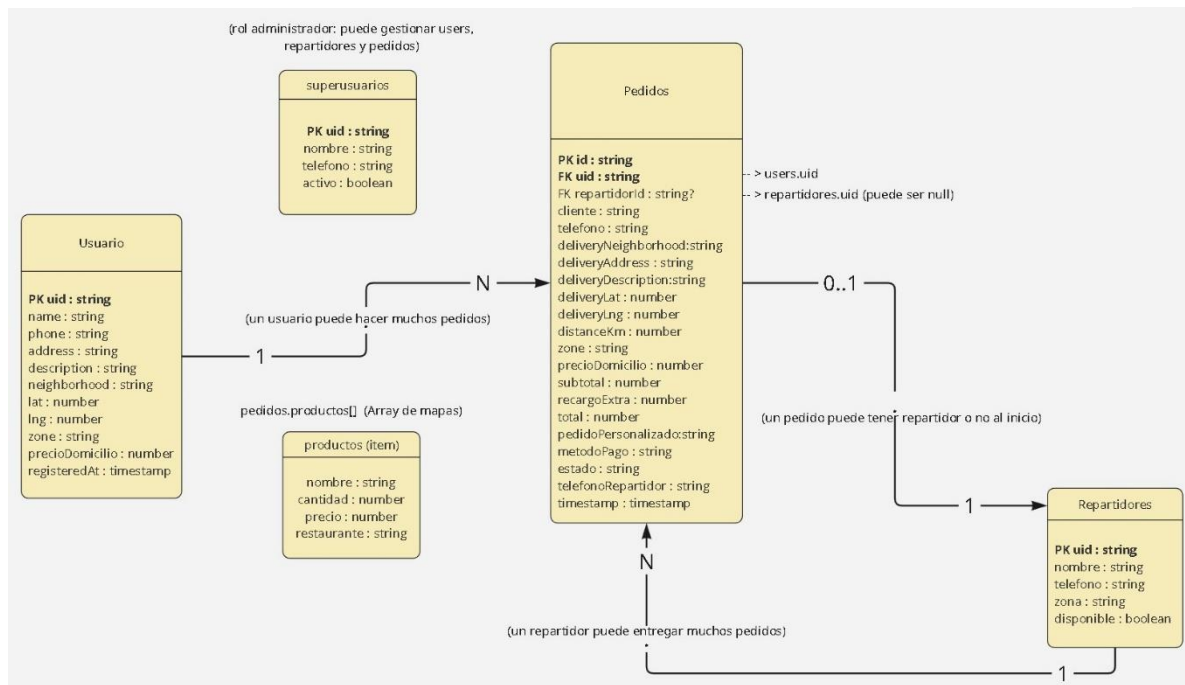
Nota* El campo productos[] corresponde al detalle del pedido y se almacena como un arreglo de mapas dentro del documento. Permite conservar una “fotografía” del contenido exacto del pedido, incluyendo nombre, cantidad, precio y restaurante.

Diagrama UML – Modelo de Colecciones y relaciones de Datos en Firestore

El diagrama UML presenta la estructura general de la base de datos implementada en Firebase Firestore para AI-instante App. En él se representan las colecciones principales del sistema: users, repartidores, superusuarios y pedidos, junto con sus campos más relevantes.

La colección pedidos actúa como eje central, ya que almacena la información completa de cada orden y mantiene relaciones lógicas mediante identificadores como uid (cliente) y repartidorId (repartidor asignado). Además, el detalle del pedido se almacena dentro del mismo documento a través del campo productos[], lo que permite conservar la información exacta de los productos solicitados en el momento de la compra.

Figura 21. Diagrama UML de colecciones y relaciones



Fuente: Elaboración Propia

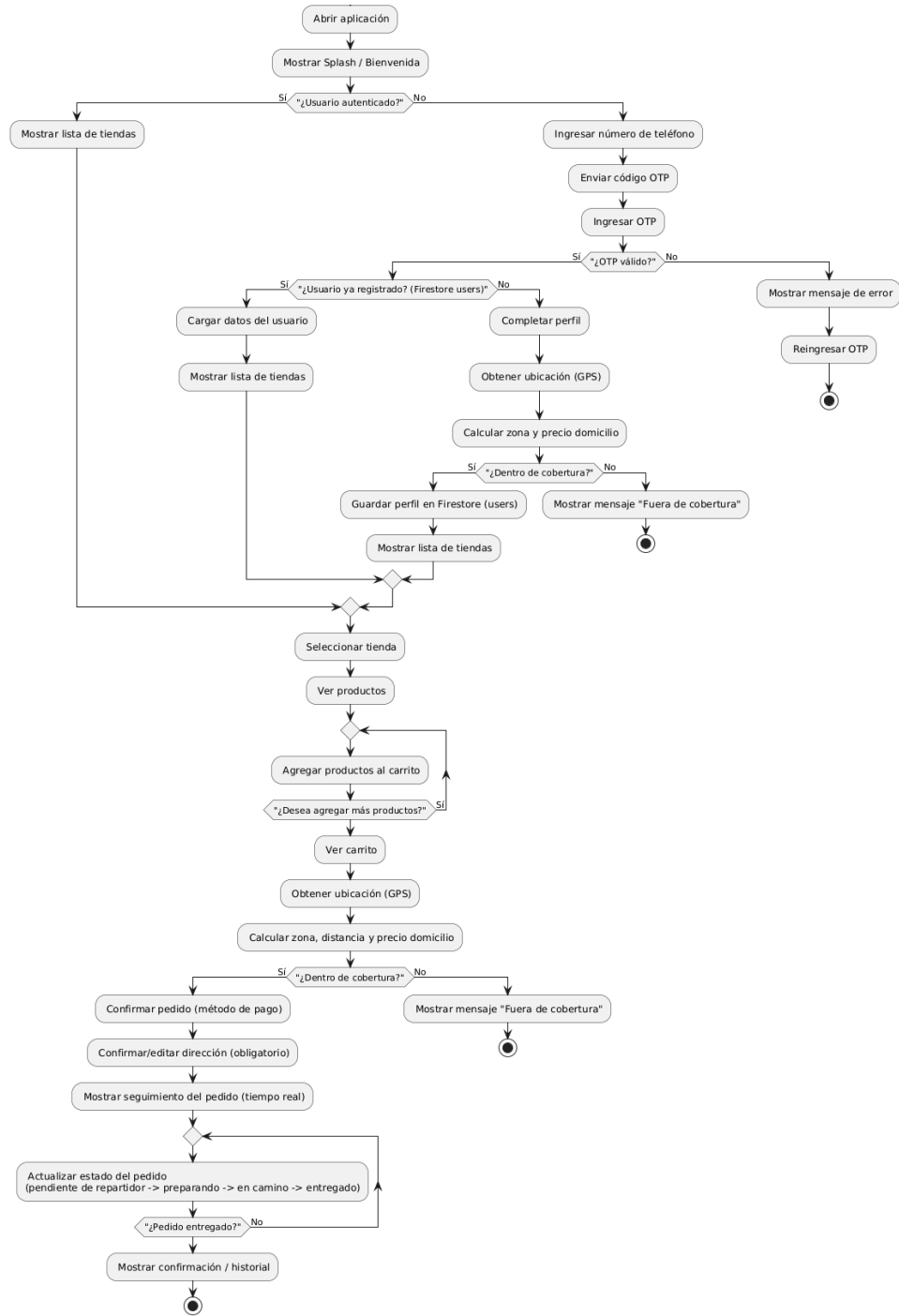
Diagrama de Actividades - Proceso General del Usuario

El diagrama de actividades representa el flujo general de uso de Al-instante App desde que el usuario abre la aplicación hasta la finalización del pedido. En este proceso se incluye la autenticación mediante número telefónico con verificación OTP, el registro del usuario en caso de ser nuevo, la navegación por los negocios disponibles, la selección de productos y la confirmación del pedido.

Asimismo, se contempla la validación de cobertura mediante ubicación GPS, el cálculo automático de la zona y costo del domicilio, y el seguimiento del estado del pedido en tiempo real hasta su entrega.

Figura 22. Diagrama de Actividades - Proceso General del

Usuario

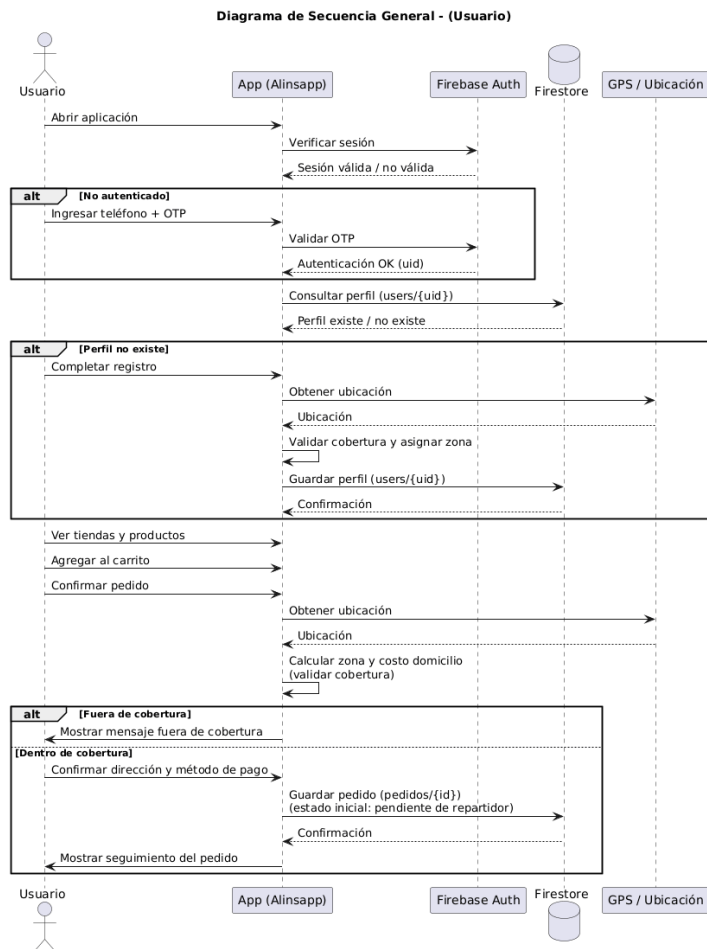


Fuente: Elaboración Propia

Diagrama de Secuencia – Funcionamiento General del Sistema

El diagrama de secuencia representa, de forma general, la interacción entre el usuario, la aplicación AI-instante App y los servicios de Firebase durante el uso del sistema. En él se describe el proceso desde la autenticación mediante OTP, la validación y registro del perfil en Firestore, hasta la creación y envío del pedido. Asimismo, se incluye la consulta de datos, la validación de cobertura mediante ubicación GPS y el almacenamiento del pedido en la base de datos, permitiendo posteriormente su seguimiento en tiempo real.

Figura 23. Diagrama de Secuencia – Funcionamiento General



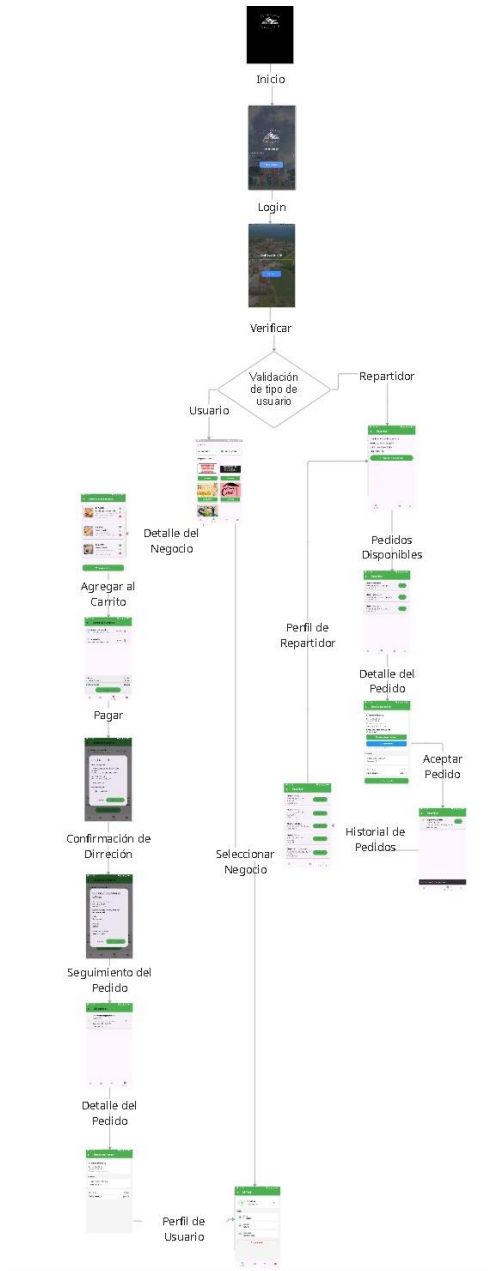
Fuente: Elaboración Propia

Diseño de Navegación

El diseño de navegación de AI-Instante App define la estructura de interacción entre las diferentes pantallas del sistema, estableciendo los flujos de acceso según el rol del usuario (cliente o repartidor). La arquitectura implementada es de tipo jerárquica y condicional, ya que el acceso a los módulos principales depende del proceso de autenticación y validación en base de datos.

El siguiente mapa de navegación representa gráficamente la organización de pantallas, las decisiones de redirección por rol y la secuencia de estados del pedido dentro del proceso operativo, garantizando coherencia funcional y trazabilidad en la experiencia de usuario.

Figura 24. Mapa de navegación del módulo de usuario y repartidor.



Fuente: Elaboración Propia

Enlace al prototipo interactivo:

https://miro.com/app/board/uXjVJRyBE_E=?moveToWidget=3458764660627738556&cot=14

Nota* Prototipo interactivo disponible bajo acceso institucional.

Fase de Programación

En la fase de programación se materializó el diseño conceptual del sistema mediante la implementación técnica de cada uno de los módulos definidos en etapas anteriores. Esta fase implicó la construcción estructurada de la aplicación móvil, la integración con servicios en la nube y la configuración de los mecanismos necesarios para garantizar seguridad, persistencia de datos y sincronización en tiempo real.

El desarrollo se realizó bajo un enfoque modular, separando responsabilidades entre componentes de autenticación, gestión de pedidos, administración de estados y validación de cobertura. Se implementó la lógica de negocio correspondiente a la autenticación por número telefónico, el registro y gestión de perfiles, el cálculo de precios de domicilio según zona, y el ciclo de vida del pedido (pendiente de repartidor, preparando, en ruta y entregado).

Asimismo, se integraron servicios backend que permitieron almacenar información estructurada, validar accesos y mantener actualizaciones automáticas entre los distintos actores del sistema (cliente y repartidor). La arquitectura adoptada permitió que los

cambios en los datos se reflejaran de manera inmediata en la interfaz del usuario, garantizando coherencia operativa y trazabilidad del proceso de entrega.

Esta fase consolidó la funcionalidad completa de Al-instante App, asegurando que cada componente implementado respondiera a los requisitos funcionales y reglas de negocio previamente establecidos.

Diseño de Seguridad

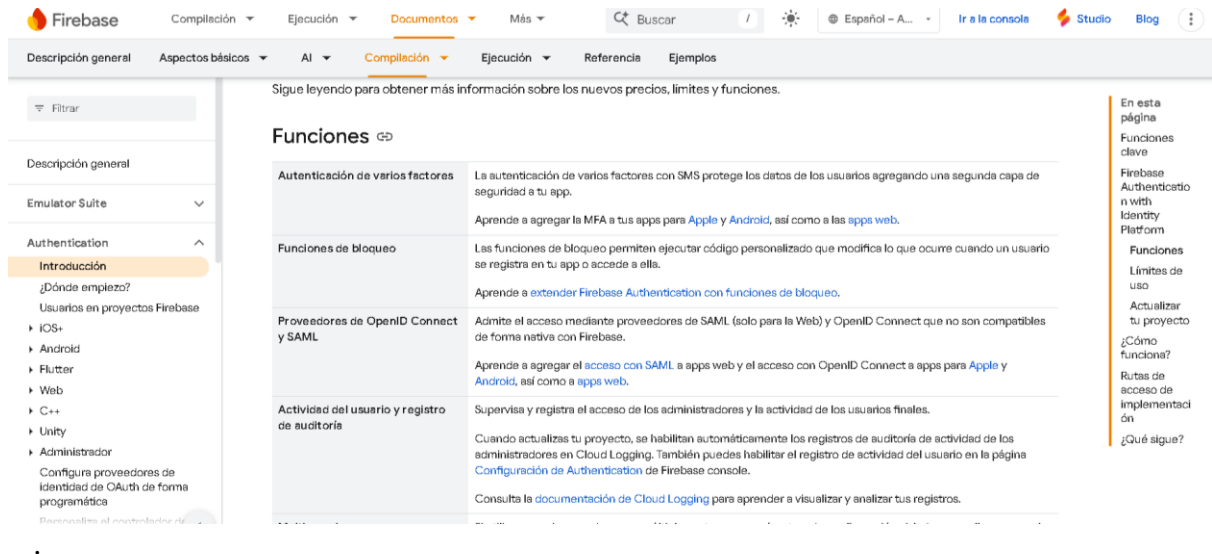
El diseño de seguridad garantizar la confidencialidad, integridad y disponibilidad de la información gestionada por Al-Instante App, protegiendo los datos de usuarios, pedidos y repartidores frente a accesos no autorizados o modificaciones indebidas.

Autenticación y control de acceso

La aplicación implementa Firebase Authentication mediante verificación por número telefónico (OTP), lo que permite validar la identidad del usuario antes de otorgar acceso al sistema. Una vez autenticado, Firebase genera un identificador único (uid) que se utiliza como referencia para controlar el acceso a la información almacenada en Firestore.

El sistema aplica el principio de mínimo privilegio, asegurando que cada usuario únicamente pueda acceder y modificar la información que le corresponde según su rol (cliente o repartidor).

Figura 25. Panel de funciones de autenticación en Firebase Console.



Fuente: Captura de pantalla de Firebase Console (2026).

Reglas de seguridad en la base de datos

La protección de los datos se refuerza mediante reglas de seguridad en Firestore, las cuales restringen la lectura y escritura de información únicamente a usuarios autenticados y autorizados.

Estas reglas garantizan que:

- Los usuarios solo puedan acceder a sus propios datos.
- Los pedidos solo sean visibles para el cliente que los creó y el repartidor asignado.
- La actualización del estado de un pedido sea realizada exclusivamente por el repartidor correspondiente.
- Se validen los campos obligatorios y los tipos de datos antes de permitir su almacenamiento.

De esta manera, la seguridad no depende únicamente de la aplicación cliente, sino que se encuentra reforzada directamente en el servidor.

Protección de datos personales

Los datos personales (nombre, teléfono y dirección) son tratados como información sensible. La aplicación adopta el principio de minimización de datos, almacenando únicamente la información necesaria para la operación del servicio.

Además, la comunicación entre la aplicación y los servidores de Firebase se realiza mediante protocolos seguros (HTTPS/TLS), garantizando el cifrado de la información durante la transmisión.

Figura 26. Reglas de seguridad implementadas en Cloud Firestore.

```
1  rules_version = '2';
2  service cloud.firestore {
3    match /databases/{database}/documents {
4
5      match /users/{userId} {
6        allow read, write: if request.auth != null
7                               && request.auth.uid == userId;
8      }
9
10     match /repartidores/{repartidorId} {
11       allow read, write: if request.auth != null
12                              && request.auth.uid == repartidorId;
13     }
14
15     match /pedidos/{pedidoId} {
16       allow create: if request.auth != null;
17
18       allow read: if request.auth != null
19                    && (resource.data.uid == request.auth.uid
20                       || resource.data.repartidorId == request.auth.uid);
21
22       allow update: if request.auth != null
23                     && resource.data.repartidorId == request.auth.uid;
24     }
25   }
26 }
27
28
```

Fuente: Elaboración Propia

Monitoreo y control

Se utiliza la consola de Firebase para supervisar accesos, errores de permisos y actividad general del sistema. Asimismo, el registro de fechas y estados de los pedidos

permite mantener trazabilidad y control sobre las operaciones realizadas dentro de la aplicación.

Herramientas y Tecnologías Utilizadas

3.1. Entorno de Desarrollo

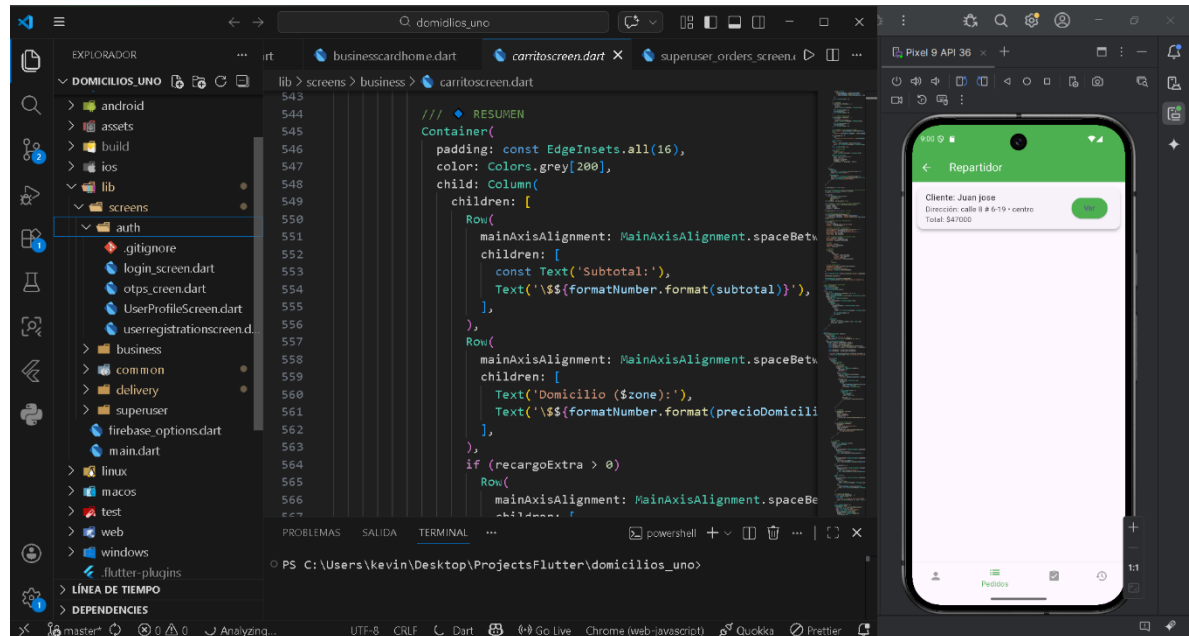
Para la construcción de AI-instante App se utilizaron entornos de desarrollo que permitieron programar, ejecutar y probar la aplicación en dispositivos móviles Android.

El editor principal fue Visual Studio Code, debido a su ligereza, rapidez y compatibilidad con extensiones oficiales de Flutter y Dart. Este entorno permitió escribir código, depurar errores, gestionar dependencias y ejecutar la aplicación directamente desde el terminal integrado.

Como complemento, se utilizó Android Studio, principalmente para la ejecución de emuladores virtuales y pruebas en distintos dispositivos Android. Esto permitió validar el comportamiento de la aplicación en diferentes tamaños de pantalla y configuraciones del sistema operativo.

Figura 27. Vista del desarrollo en Visual Studio Code con ejecución en emulador

Android



Fuente: Elaboración Propia

3.2. Lenguaje de Programación: Dart

El desarrollo de la aplicación se realizó utilizando Dart, lenguaje de programación creado por Google y diseñado para la construcción de aplicaciones modernas.

Dart es un lenguaje orientado a objetos, con tipado fuerte y sintaxis clara, lo que facilita la organización del código y la implementación de estructuras modulares. En el proyecto, Dart permitió desarrollar:

- La lógica de negocio (validación de cobertura, cálculo de precios, gestión de estados).
- La interacción con servicios en la nube.
- La estructura visual de cada pantalla.

Su integración nativa con Flutter permitió una construcción eficiente tanto de la interfaz gráfica como del comportamiento interno de la aplicación.

Figura 28. Documentación oficial del lenguaje de programación Dart.



Fuente: Página oficial de documentación dart.dev.

3.3. Framework de Desarrollo: Flutter

El framework utilizado fue Flutter, una tecnología multiplataforma que permite desarrollar aplicaciones móviles con una única base de código.

Flutter funciona mediante un sistema basado en widgets, donde cada componente visual (botones, textos, listas, formularios) se construye como un elemento independiente y reutilizable. Esto facilitó el diseño de pantallas como:

- Autenticación por número telefónico.
- Listado de tiendas.
- Carrito de compras.
- Seguimiento de pedidos por estados.

Su arquitectura reactiva permitió actualizar automáticamente la interfaz cuando cambian los datos, por ejemplo, cuando el estado del pedido pasa de “pendiente de repartidor” a “preparando”.

Figura 29. Logo oficial de Flutter.



Fuente: Página oficial Flutter.dev.

3.4. Backend y Servicios en la Nube: Firebase

Para la gestión de datos y autenticación se utilizó Firebase, plataforma de servicios en la nube que permitió implementar un backend sin necesidad de configurar servidores propios.

3.4.1. Firebase Authentication

Se implementó el método de autenticación mediante número telefónico con verificación por código OTP. Este mecanismo garantiza seguridad en el acceso y simplifica la experiencia del usuario al no requerir contraseñas tradicionales.

3.4.2. Cloud Firestore

Cloud Firestore se utilizó como base de datos NoSQL en tiempo real. En ella se estructuraron las colecciones principales del sistema:

- users
- repartidores
- pedidos

Firestore permitió la sincronización automática de datos, de modo que cuando el repartidor actualiza el estado del pedido (pendiente de repartidor - preparando - en ruta- entregado), el usuario puede visualizar el cambio de manera inmediata.

3.4.3 Firebase Cloud Messaging (FCM)

Se integró el servicio de notificaciones para alertar a los repartidores cuando existe un nuevo pedido disponible. Esto mejora la eficiencia operativa y reduce tiempos de respuesta.

3.5. Control de Versiones: Git y GitHub

Para el seguimiento del desarrollo se empleó Git como sistema de control de versiones, permitiendo registrar cada modificación realizada en el código fuente.

Asimismo, se utilizó GitHub como repositorio remoto, lo que facilitó el almacenamiento seguro del proyecto, la recuperación de versiones anteriores y la documentación del avance mediante historial de commits.

Evidencia de trazabilidad

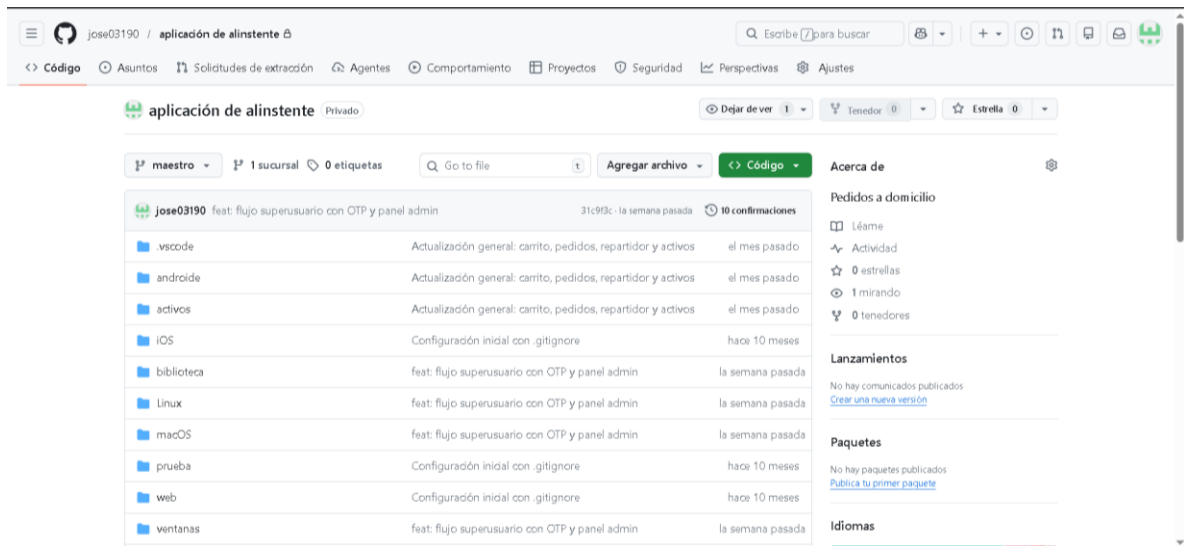
La fase de construcción del sistema fue gestionada mediante control de versiones utilizando GitHub, permitiendo mantener un registro cronológico verificable de cada implementación realizada.

El repositorio evidencia actividad desde abril de 2025, donde se realizó la configuración inicial del proyecto y la implementación de funcionalidades base como la gestión de estados de pedidos (preparando, en ruta, entregado), integración de UID y sincronización entre usuario y repartidor.

Posteriormente, durante diciembre de 2025 y enero–febrero de 2026, se consolidaron funcionalidades críticas como la implementación del método de pago, correcciones en la persistencia de datos en Firestore, mejoras en el carrito de compras, optimización del flujo de estados y desarrollo del panel de superusuario con autenticación OTP.

Las fechas distribuidas en el historial de commits responden a un proceso de desarrollo incremental y mejora continua, propio de metodologías iterativas de ingeniería de software. Esto garantiza la trazabilidad técnica del proyecto y demuestra su evolución progresiva dentro del periodo académico.

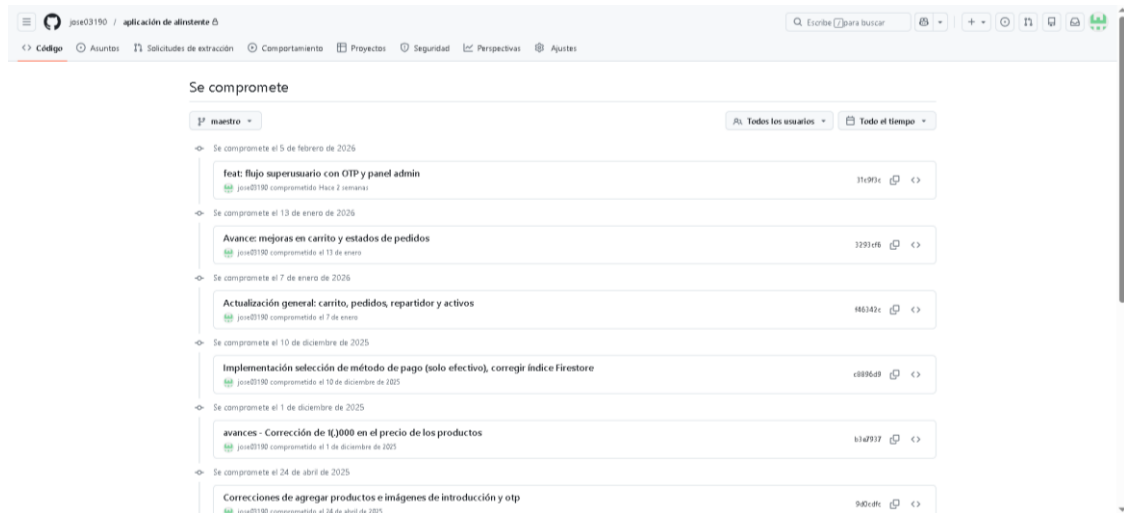
Figura 30. Repositorio del proyecto Al-instante App en GitHub (rama principal y estructura de carpetas).



Fuente: Elaboración Propia

Esta sección muestra la vista principal del repositorio del proyecto en GitHub, donde se almacena el código fuente y se gestiona el control de versiones. Desde esta interfaz se visualiza la rama principal del proyecto, la estructura de carpetas y el último cambio registrado, permitiendo mantener un respaldo en la nube y un historial verificable del desarrollo.

Figura 31. *Historial de confirmaciones (commits) del repositorio en GitHub.*



Fuente: Elaboración Propia

Tabla 17. *Tabla metodológica basada en tus commits*

Fecha	Commit	Funcionalidad implementada	Fase metodológica
Abril 2025	Configuración inicial	Creación estructura del proyecto	Inicio / Prototipado
Abril 2025	Estados (preparando, en ruta, entregado)	Lógica de pedidos	Construcción
Abril 2025	Sincronización usuario–repartidor	Integración de módulos	Construcción
Dic 2025	Método de pago + Firestore	Persistencia y validación	Construcción
Ene 2026	Mejoras carrito y estados	Optimización funcional	Refinamiento
Feb 2026	Panel superusuario	Ampliación del sistema	Mejora evolutiva

Nota* Datos obtenidos del historial de versiones del repositorio oficial del proyecto en GitHub.

README TÉCNICO DEL PROYECTO

Sistema: al_instante_domicilios

1. Información General del Proyecto

Nombre del proyecto: al_instante_domicilios

Tipo de aplicación: Aplicación móvil Android para gestión de domicilios.

Lenguaje de programación: Dart

Framework principal: Flutter

Gestión de estado: Provider

Base de datos: Cloud Firestore (Firebase)

Autenticación: Firebase Authentication (verificación por número telefónico mediante SMS)

Repositorio: GitHub (repositorio privado)

La aplicación fue desarrollada bajo una arquitectura modular organizada por dominios funcionales, permitiendo escalabilidad, mantenibilidad y claridad estructural en el desarrollo.

A.1 Versiones del Entorno de Desarrollo

El sistema **al_instante_domicilios** fue desarrollado bajo las siguientes versiones oficiales del entorno tecnológico:

- Flutter 3.29.2 (canal estable)
- Dart SDK 3.7.2
- DevTools 2.42.3

- Sistema operativo de desarrollo: Windows x64

Estas versiones garantizan compatibilidad con las dependencias implementadas y permiten la correcta reproducción del entorno de desarrollo.

Figura 32. Configuración del entorno de desarrollo Flutter.

```
Flutter 3.29.2 • channel stable • https://github.com/flutter/flutter.git
Framework • revision c236373904 (12 months ago) • 2025-03-13 16:17:06 -0400
Engine • revision 18b71d647a
Tools • Dart 3.7.2 • DevTools 2.42.3
Dart SDK version: 3.7.2 (stable) (Tue Mar 11 04:27:50 2025 -0700) on "windows_x64"
```

Fuente: Elaboración Propia

A.2 Estructura del Proyecto

El proyecto adopta una arquitectura modular organizada por dominios funcionales dentro del directorio lib/, permitiendo separación clara de responsabilidades.

La estructura principal se compone de:

- Archivo principal de ejecución (main.dart)
- Archivo de configuración de Firebase (firebase_options.dart)
- Carpeta screens/, subdividida en módulos funcionales

A.2.1 Organización por Módulos

Módulo auth/

Gestiona la autenticación mediante número telefónico (OTP) y el registro de usuarios.

Módulo business/

Contiene la lógica relacionada con restaurantes, productos, carrito global y gestión de pedidos.

Módulo delivery/

Implementa las funcionalidades del repartidor: aceptación de pedidos, actualización de estados e historial.

Módulo superuser/

Permite la administración de negocios, repartidores y pedidos.

Módulo common/

Incluye componentes compartidos del sistema, como ruteo por rol y pantalla de carga inicial.

Figura 33. Imagen del árbol del proyecto exportado desde la terminal

```
PS C:\Users\kevin\Desktop\ProjectsFlutter\domicilios_uno> tree lib /F /A
lib
├── firebase_options.dart
├── main.dart
├── screens
│   ├── auth
│   │   ├── .gitignore
│   │   ├── login_screen.dart
│   │   ├── otps_screen.dart
│   │   ├── UserProfileScreen.dart
│   │   └── userregistrationscreen.dart
│   ├── business
│   │   ├── businesscard.dart
│   │   ├── businesscardhome.dart
│   │   ├── business_screen.dart
│   │   ├── carritoscreen.dart
│   │   ├── cart_provider.dart
│   │   ├── estanquillo_vina_screen.dart
│   │   ├── hiernos_palestina_screen.dart
│   │   ├── kikas_pizza_screen.dart
│   │   ├── mandados_screen.dart
│   │   ├── mirador_colina_screen.dart
│   │   ├── panfilla_artesanal_screen.dart
│   │   ├── sandra_comidas_screen.dart
│   │   ├── user_orders_screen.dart
│   │   └── user_order_detail_view.dart
│   ├── estanquillo
│   │   ├── estanquillo_products.dart
│   │   └── estanquillo_screen.dart
│   └── widgets
│       ├── section_header.dart
│       └── service_card.dart
├── common
│   ├── app_colors.dart
│   ├── role_router_screen.dart
│   ├── splashscreen.dart
│   └── zone_service.dart
├── delivery
│   ├── delivery_assigned_orders.dart
│   ├── delivery_history.dart
│   ├── delivery_home.dart
│   ├── delivery_login.dart
│   ├── delivery_order_detail.dart
│   ├── delivery_order_detail_view.dart
│   ├── delivery_order_list.dart
│   └── delivery_registration.dart
└── superuser
    ├── superuser_businesses_screen.dart
    ├── superuser_couriers_screen.dart
    ├── superuser_home_screen.dart
    └── superuser_orders_screen.dart
```

Fuente: Elaboración Propia

A.3 Estructura Arquitectónica

El sistema se encuentra estructurado bajo una organización por capas funcionales:

- **Capa de presentación:** Interfaces desarrolladas en Flutter (pantallas y widgets).

- **Gestión de estado:** Implementada mediante el patrón Provider para el carrito global.
- **Capa de servicios:** Integración con Firebase Authentication y Cloud Firestore.
- **Control por roles:** Diferenciación entre usuario, repartidor y superusuario mediante ruteo dinámico.

Esta arquitectura facilita la mantenibilidad, escalabilidad y claridad del sistema.

A.4 Configuración de Firebase

El proyecto está integrado con Firebase mediante FlutterFire CLI, generando el archivo de configuración:

firebase_options.dart

El sistema utiliza los siguientes servicios:

- Firebase Authentication (verificación OTP por SMS)
- Cloud Firestore como base de datos NoSQL

Para la ejecución en Android se requiere el archivo:

android/app/google-services.json

Por motivos de seguridad, las credenciales, claves API y tokens de autenticación no se incluyen en este documento.

Las principales colecciones implementadas en Firestore son:

- usuarios
- repartidores
- pedidos

A.5 Comandos de Ejecución

Para la correcta instalación y ejecución del proyecto se deben seguir los siguientes pasos:

1. Clonar el repositorio privado del proyecto.
2. Acceder al directorio raíz del sistema.
3. Ejecutar la instalación de dependencias mediante flutter pub get.
4. Verificar el entorno con flutter doctor.
5. Ejecutar la aplicación mediante flutter run.

El sistema actualmente se encuentra configurado para su ejecución en entorno Android.

Figura 34. Ejecución de comandos para la configuración y prueba del sistema en entorno Flutter.

```
> flutter pub get
>> flutter doctor
>> flutter run
```

Fuente: Elaboración Propia

Es requisito contar con el archivo de configuración de Firebase (google-services.json) ubicado en android/app/.

Conclusiones de la Fase de Programación

La fase de programación permitió transformar el diseño conceptual de AI-instante App en una solución funcional, integrando de manera coherente los distintos módulos del

sistema y garantizando su operatividad en un entorno real. A través de una arquitectura modular y el uso de tecnologías modernas, se logró implementar una aplicación estructurada, escalable y alineada con los requisitos definidos en la etapa de análisis.

La integración entre Flutter y Firebase demostró ser adecuada para el desarrollo de una plataforma con sincronización en tiempo real, permitiendo que los cambios en los estados de los pedidos se reflejaran de manera inmediata entre cliente y repartidor. Esto aseguró coherencia operativa, trazabilidad en el proceso de entrega y una experiencia de usuario fluida.

Asimismo, la implementación de autenticación mediante OTP, la gestión estructurada de datos en Cloud Firestore y el uso de notificaciones contribuyeron a fortalecer la seguridad, disponibilidad y confiabilidad del sistema. El manejo adecuado del control de versiones mediante Git y GitHub garantizó respaldo del código, trazabilidad del desarrollo y organización en las modificaciones realizadas.

En términos generales, esta fase consolidó técnicamente la aplicación, permitiendo que cada funcionalidad implementada respondiera de forma efectiva a las reglas de negocio establecidas, sentando una base sólida para la etapa de pruebas y validación del sistema.

Fase de Pruebas

La fase de pruebas tuvo como propósito validar el correcto funcionamiento de la aplicación y evaluar su comportamiento en condiciones reales de uso. Se realizaron pruebas funcionales orientadas a verificar el cumplimiento de los requerimientos establecidos, así como pruebas de usabilidad para evaluar la experiencia del usuario.

Asimismo, se llevó a cabo una prueba piloto con usuarios seleccionados, quienes interactuaron con la aplicación y proporcionaron retroalimentación sobre su desempeño, facilidad de uso y utilidad general. Esta información permitió identificar oportunidades de mejora y realizar ajustes finales antes de considerar la aplicación como una solución funcional.

Pruebas y Validación de la Solución

1 Estrategia de pruebas

La estrategia de pruebas aplicada en el desarrollo de la aplicación AI-Instante App fue de tipo funcional, basada en la validación de cada requisito funcional definido en el proyecto.

Se ejecutaron pruebas manuales estructuradas bajo los siguientes criterios:

- Validación de autenticación mediante OTP.
- Verificación de redirección por rol (cliente, repartidor, administrador).
- Pruebas de creación, asignación y actualización de pedidos.
- Validación de persistencia en base de datos (Firestore).
- Pruebas de sincronización en tiempo real (StreamBuilder).
- Validación del módulo administrativo (Super Usuario).
- Registro de incidencias y corrección de errores detectados.

Las pruebas se ejecutaron en entorno real utilizando dispositivos móviles con conexión activa a Firebase.

Tabla 18. Matriz de Trazabilidad

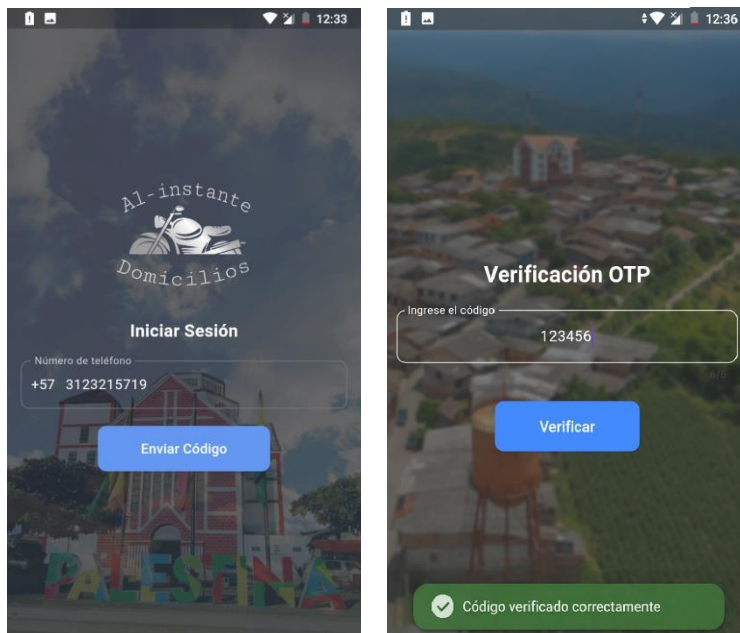
Requisito (RF)	Prueba (ID)	Resultado
RF-01 Autenticación OTP	TP-01	Aprobado
RF-01 Identificación de Rol	TP-03	Aprobado
RF-02 Registro de Usuario	TP-04	Aprobado
RF-03 Gestión Admin – Repartidores	TP-05A	Aprobado
RF-03 Gestión Admin – Negocios	TP-05B	Aprobado
RF-04 Realizar Pedido	TP-06	Aprobado
RF-05 Asignación Repartidor	TP-07	Aprobado
RF-06 Actualización Estado	TP-08 / TP-09 / TP-10	Aprobado
RF-07 Método de Pago	TP-11	Aprobado
RF-08 Seguimiento Pedido Cliente	TP-12	Aprobado
RF-09 Pedidos Disponibles	TP-13	Aprobado
RF-10 Monitoreo Pedidos Admin	TP-14	Aprobado
RF-11 Pedidos Asignados	TP-15	Aprobado

2. Plan de Pruebas

Tabla 19. TP-01 – Autenticación con OTP (RF-01)

Campo	Descripción
ID prueba	TP-01
Requisito asociado	RF-01
Pasos	Ingresar número - recibir OTP - ingresar OTP correcto
Datos de prueba	Número real + código SMS
Resultado esperado	Acceso correcto al sistema
Resultado obtenido	Aprobado
Evidencia	Captura Login y OTP

Figura 35. Captura Login y OTP.

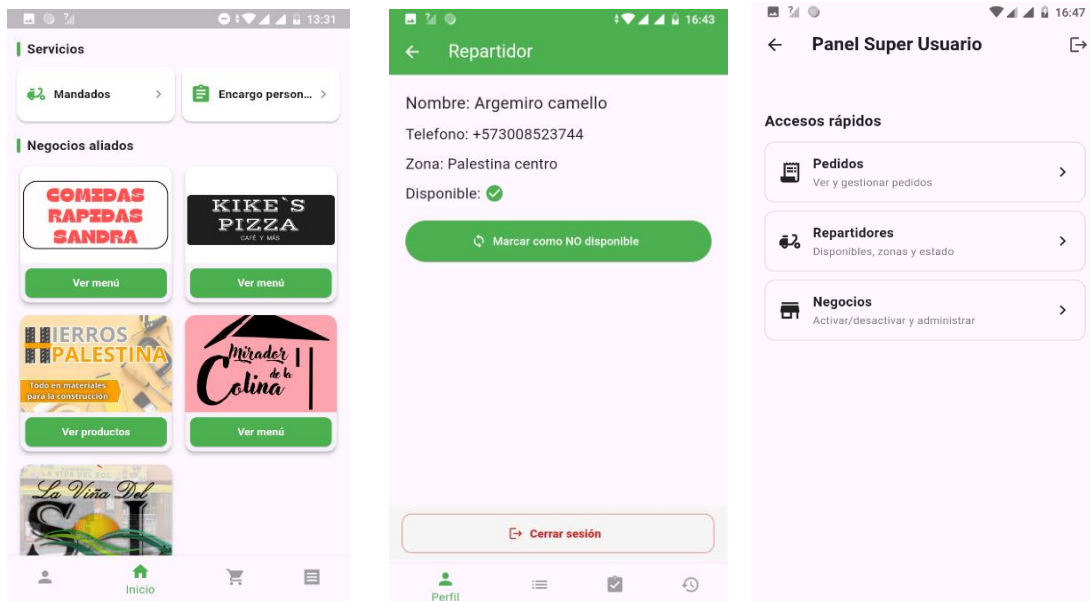


Fuente: Elaboración Propia

Tabla 20. TP-03 – Identificación de Rol (RF-01)

Campo	Descripción
ID prueba	TP-03
Requisito asociado	RF-01
Pasos	Iniciar sesión con número de cliente, repartidor y admin
Datos de prueba	3 números registrados con roles distintos
Resultado esperado	Redirección al Home de inicio correspondiente según rol
Resultado obtenido	Aprobado
Evidencia	Captura Home Cliente, Inicio Repartidor, Panel Super Usuario

Figura 36. Captura Home Cliente, Inicio Repartidor, Panel Super Usuario.

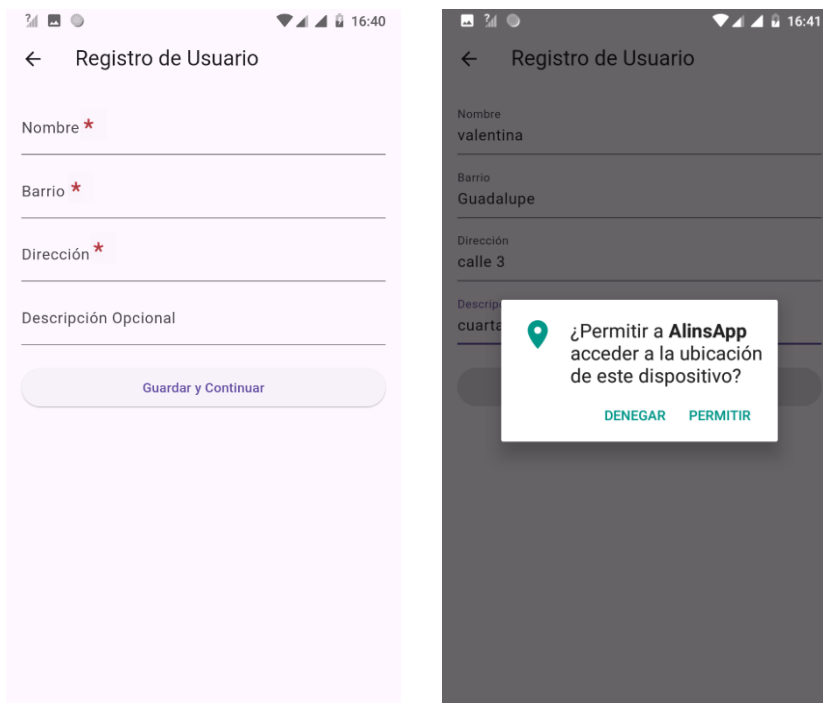


Fuente: Elaboración Propia

Tabla 21. TP-04 – Registro de Usuario (RF-02)

Campo	Descripción
ID prueba	TP-04
Requisito asociado	RF-02
Pasos	Completar formulario en UserRegistrationScreen - Guardar
Datos de prueba	Nombre, barrio, dirección, descripción
Resultado esperado	Registro en colección users y redirección a Home
Resultado obtenido	Aprobado
Evidencia	Captura formulario de registro

Figura 37. Captura formulario de registro.

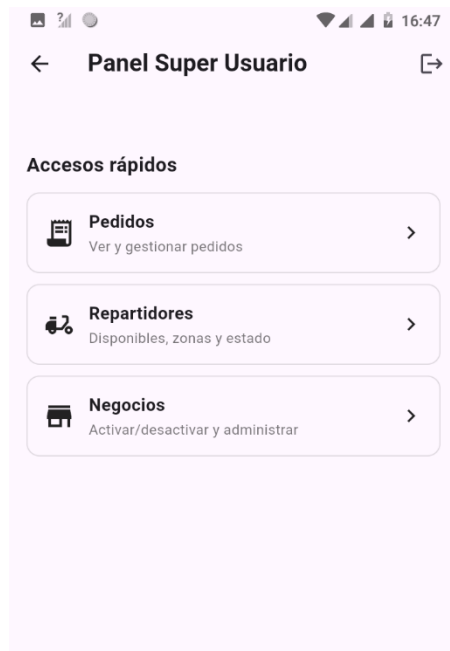


Fuente: Elaboración Propia

Tabla 22. TP-05A – Gestión Repartidores Admin (RF-03)

Campo	Descripción
ID prueba	TP-05A
Requisito asociado	RF-03
Pasos	Admin OTP - Panel Super Usuario - Repartidores
Datos de prueba	Repartidores registrados
Resultado esperado	Visualización y gestión de repartidores
Resultado obtenido	Aprobado
Evidencia	Captura panel admin y lista repartidores

Figura 38. Captura panel admin, lista de pedidos, repartidores y negocios.



Fuente: Elaboración Propia

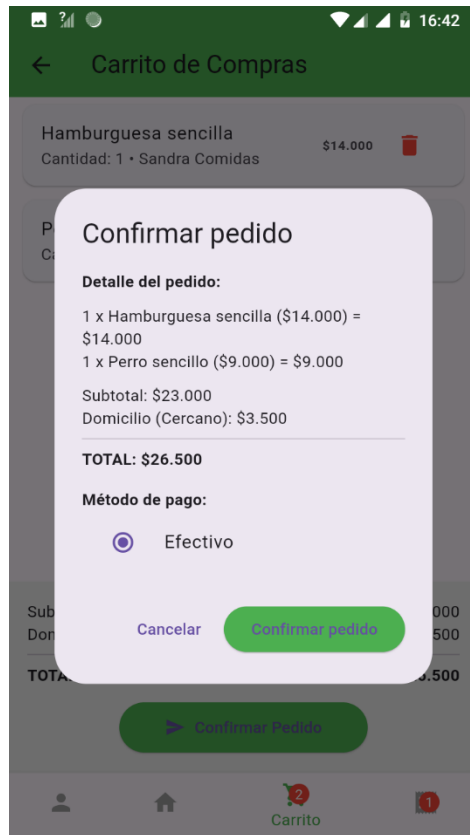
Tabla 23. TP-05B – Gestión Negocios Admin (RF-03)

Campo	Descripción
ID prueba	TP-05B
Requisito asociado	RF-03
Pasos	Admin OTP - Panel Super Usuario - Negocios
Datos de prueba	Negocios registrados
Resultado esperado	Administración de negocios
Resultado obtenido	Aprobado
Evidencia	Captura panel admin y negocios

Tabla 24. TP-06 – Realizar Pedido (RF-04)

Campo	Descripción
ID prueba	TP-06
Requisito asociado	RF-04
Pasos	Agregar productos - Confirmar - Guardar pedido
Datos de prueba	Productos del carrito
Resultado esperado	Pedido guardado con estado “pendiente de repartidor”
Resultado obtenido	Aprobado
Evidencia	Captura carrito y confirmación

Figura 39. Captura carrito y confirmación.



Fuente: Elaboración Propia

Tabla 25. TP-07 – Asignación de Repartidor (RF-05)

Campo	Descripción
ID prueba	TP-07
Requisito asociado	RF-05
Pasos	Repartidor acepta - pedido pendiente
Datos de prueba	Pedido en estado “pendiente de repartidor”

Campo	Descripción
Resultado esperado	Pedido asignado al repartidor
Resultado obtenido	Aprobado
Evidencia	Captura lista repartidor

Figura 40. Captura lista repartidor.



Fuente: Elaboración Propia

Tabla 26. TP-08 / TP-09 / TP-10 – Actualización de Estado (RF-06)

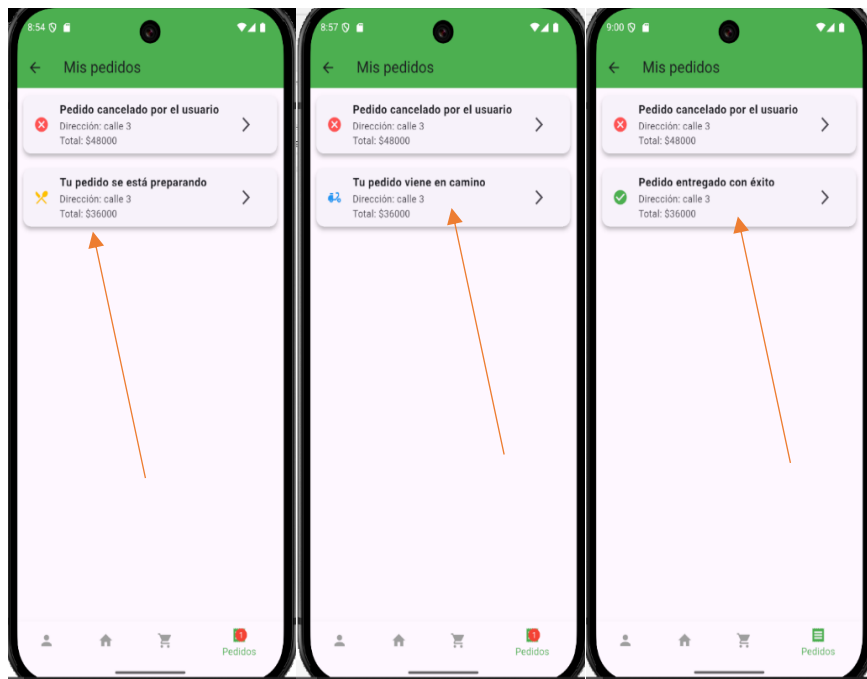
Estado probado	Resultado
preparando	Aprobado
en ruta	Aprobado

Estado probado Resultado

entregado Aprobado

Evidencia Capturas de cambio de estados.

Figura 41. Capturas de cambio de estados.



Fuente: Elaboración Propia

Tabla 27. TP-11 – Método de Pago (RF-07)

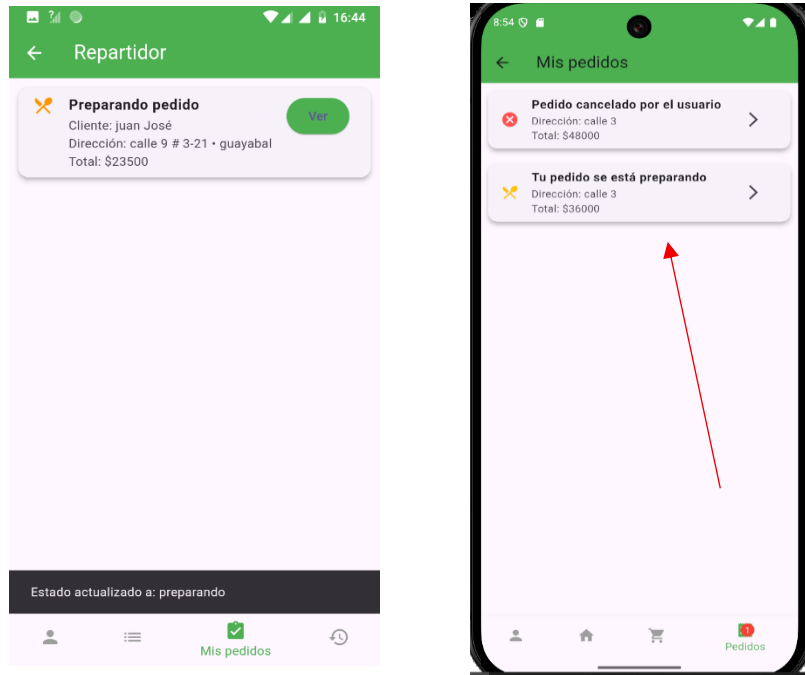
Campo	Descripción
ID prueba	TP-11
Requisito asociado	RF-07
Pasos	Confirmar pedido con método “efectivo”

Campo	Descripción
Datos de prueba	metodoPago = "efectivo"
Resultado esperado	Guardado correcto en Firestore
Resultado obtenido	Aprobado
Evidencia	Captura confirmación.

Tabla 28. *TP-12 – Seguimiento Pedido Cliente (RF-08)*

Campo	Descripción
ID prueba	TP-12
Requisito asociado	RF-08
Pasos	Cliente abre Mis pedidos - Repartidor cambia estado
Datos de prueba	Pedido activo
Resultado esperado	Actualización en tiempo real
Resultado obtenido	Aprobado
Evidencia	Captura antes/después

Figura 42. Captura antes/después, repartidor y usuario.



Fuente: Elaboración Propia

Resultados

1. Resultados medibles

Con el fin de validar el funcionamiento general de la aplicación AI-instante App, se definió un plan de pruebas compuesto por 14 casos. De estos, los 14 fueron ejecutados satisfactoriamente, obteniendo un total de 14 casos aprobados y 0 casos fallidos. En

consecuencia, el porcentaje de aprobación alcanzado fue del 100%, lo que evidencia un comportamiento funcional estable del sistema durante el proceso de evaluación.

2. Pruebas funcionales

Las pruebas funcionales se orientaron a comprobar que cada una de las funcionalidades principales de Al-instante App operara conforme a los requerimientos establecidos. Para ello, se validaron procesos clave como la autenticación de usuarios por número telefónico, el registro de perfiles, la creación y gestión de pedidos, la asignación de repartidores, la actualización de estados y el envío de notificaciones.

Durante esta etapa se verificó que la información ingresada por los usuarios se almacenara correctamente en la base de datos, que los cambios de estado se reflejaran en tiempo real y que el sistema respondiera de manera adecuada ante diferentes escenarios de uso. Las pruebas permitieron detectar y corregir inconsistencias menores, mejorando la estabilidad general de la aplicación.

3. Prueba piloto con usuarios

Como parte del proceso de validación, se realizó una prueba piloto con un grupo reducido de usuarios, quienes interactuaron con la aplicación simulando el proceso real de solicitud y entrega de pedidos. Esta prueba permitió evaluar la facilidad de uso, la claridad de la interfaz y la comprensión general del flujo de la aplicación.

La retroalimentación obtenida evidenció una buena aceptación de la solución, destacándose la rapidez del proceso de pedido, la claridad en el seguimiento del estado y la utilidad de las notificaciones. Asimismo, se identificaron oportunidades de mejora relacionadas con ajustes visuales y optimización de algunos tiempos de respuesta, las cuales fueron consideradas para iteraciones posteriores.

4. Incidencias detectadas y corregidas

Durante la fase de pruebas se identificaron algunas incidencias puntuales que afectaban la experiencia de uso. No obstante, estas fueron corregidas satisfactoriamente antes de la validación final del sistema.

Tabla 29. Incidencias Detectadas y Corregidas

Código	Incidencia	Situación Inicial	Corrección Aplicada	Estado Final
BUG-01	Splash duplicado	Splash se duplicaba y demoraba	Ajuste en flujo de navegación	Corregido
BUG-02	Flecha atrás repartidor	Regresaba al login accidentalmente	Control de salida solo por botón “Cerrar sesión”	Corregido

5. Resultados obtenidos

Los resultados de las pruebas realizadas demuestran que Al-instante App cumple con los objetivos planteados en el proyecto, ofreciendo una solución funcional para la gestión de servicios de domicilios. La aplicación permitió optimizar el proceso de pedidos, mejorar la comunicación entre clientes y repartidores y centralizar la información en una plataforma digital accesible.

Desde el punto de vista técnico, el sistema mostró un comportamiento estable durante las pruebas, con una correcta sincronización de datos y una respuesta adecuada ante las acciones de los usuarios. Desde la perspectiva del usuario, la aplicación fue percibida como una herramienta práctica y fácil de utilizar.

Discusión

1. Interpretación de los resultados

Los resultados obtenidos permiten evidenciar que la solución desarrollada responde de manera favorable a la problemática identificada al inicio del proyecto. La implementación de AI-instante App demostró que es posible digitalizar y organizar el servicio de domicilios mediante una aplicación móvil que integra procesos de registro, gestión de pedidos, seguimiento de estados y comunicación entre los actores involucrados.

El cumplimiento total de los casos de prueba ejecutados refleja que las funcionalidades principales fueron implementadas de forma adecuada y que el sistema alcanzó un nivel de estabilidad satisfactorio para su validación. De igual manera, la prueba piloto mostró que la aplicación resulta comprensible y útil para los usuarios, lo cual fortalece la viabilidad de la propuesta.

2. Cumplimiento de los objetivos del proyecto

A partir de los resultados obtenidos, se puede afirmar que el proyecto logró cumplir con el objetivo general de desarrollar una aplicación móvil para la gestión del servicio de domicilios. Asimismo, se evidencia el cumplimiento de los objetivos específicos relacionados con el diseño e implementación de módulos funcionales, la validación del sistema mediante pruebas y la mejora de la interacción entre clientes y repartidores.

La aplicación no solo permitió automatizar procesos que anteriormente podían realizarse de manera manual o poco estructurada, sino que también facilitó el acceso a información actualizada sobre pedidos y estados de entrega.

3. Aportes de la solución desarrollada

Uno de los principales aportes de AI-instante App consiste en ofrecer una alternativa tecnológica orientada a optimizar la prestación del servicio de domicilios en contextos locales. La solución desarrollada centraliza la información, mejora la trazabilidad de los pedidos y fortalece la experiencia del usuario mediante notificaciones y seguimiento en tiempo real.

Además, el uso de tecnologías como Flutter y Firebase permitió construir una solución funcional, escalable y adecuada para futuras ampliaciones, lo que representa un valor agregado desde el punto de vista técnico y operativo.

4. Limitaciones y oportunidades de mejora

Aunque los resultados fueron favorables, durante el proceso de validación se identificaron algunas oportunidades de mejora relacionadas con ajustes visuales y optimización en tiempos de respuesta. Esto indica que, si bien la aplicación cumple su propósito principal, aún existen aspectos susceptibles de perfeccionamiento en futuras versiones.

En este sentido, el proyecto puede evolucionar mediante la incorporación de nuevas funcionalidades, mejoras en rendimiento, ampliación del número de usuarios en pruebas piloto y fortalecimiento de la experiencia de uso según las necesidades del entorno.

Conclusiones

1. El desarrollo de AI-instante App permitió diseñar e implementar una solución tecnológica funcional para la gestión del servicio de domicilios, respondiendo a la necesidad de optimizar los procesos de pedido, asignación y seguimiento dentro de una plataforma digital.
2. Los resultados obtenidos en las pruebas funcionales evidenciaron un desempeño satisfactorio del sistema, alcanzando un 100% de aprobación en los casos evaluados, lo cual demuestra la estabilidad y confiabilidad de las funcionalidades principales implementadas.
3. La prueba piloto realizada con usuarios permitió confirmar que la aplicación presenta una interfaz comprensible, un flujo de uso adecuado y una utilidad práctica para la gestión de pedidos, destacándose especialmente el seguimiento del estado y el uso de notificaciones.
4. La solución desarrollada contribuye a mejorar la comunicación entre clientes y repartidores, centralizar la información del servicio y fortalecer la experiencia del usuario, constituyéndose en una herramienta con potencial de aplicación en contextos locales y regionales.
5. El enfoque iterativo adoptado durante el desarrollo facilitó la detección y corrección de incidencias, así como el ajuste progresivo de la aplicación conforme a las necesidades identificadas en las pruebas y en la retroalimentación obtenida.
6. Aunque la aplicación alcanzó los objetivos propuestos, se identificaron oportunidades de mejora relacionadas con aspectos visuales, tiempos de respuesta y

ampliación de funcionalidades, lo que abre la posibilidad de futuras fases de desarrollo e investigación.

Referencias

- Alcaldía de Bogotá. (2021, marzo 19). *ABC del decreto sobre empresas de domicilios y domiciliarios*. Recuperado de <https://bogota.gov.co/mi-ciudad/gobierno/abc-del-decreto-sobre-empresas-de-domicilios-y-domiciliarios>
- Alcaldía de Palestina Caldas. (2025). Recuperado de <https://www.palestina-caldas.gov.co>
- Armadillo Amarillo. (2022). *Mejores tecnologías para desarrollo de aplicaciones móviles*. Recuperado de <https://www.armadilloamarillo.com/blog/mejores-tecnologias-para-desarrollo-de-aplicaciones-moviles/>
- Asociación Nacional de Comercio Exterior. (2021, agosto 25). *La logística en Colombia: Cómo viene y para dónde apunta*. Analdex. Recuperado de <https://analdex.org/2021/08/25/logisticaencolombia/>
- Bambú Mobile. (2024). *7 tecnologías usadas para el desarrollo de apps 2024*. Recuperado de <https://bambu-mobile.com/tecnologias-usadas-para-el-desarrollo-de-apps-2024/>
- Banco Interamericano de Desarrollo (BID). (2023). *Transición productiva y digitalización empresarial en la región andina*. Recuperado de <https://publications.iadb.org/publications/spanish/document/Transicion-productiva-y-digitalizacion-empresarial-en-la-region-andina.pdf>
- Banco Mundial. (2023). *Economía Digital para América Latina y el Caribe: Diagnóstico de país – Colombia*. Recuperado de <https://www.bancomundial.org/es/programs/de4lac/publication/digital-economy-for-latin-america-and-the-caribbean-country-diagnostic-colombia>
- Bloomberg Línea. (2021). *¿Cuántas personas trabajan con apps de domicilios y movilidad en Colombia?* Recuperado de <https://www.bloomberglinea.com/2021/09/22/cuantas-personas-trabajan-con-apps-de-domicilios-y-movilidad-en-colombia/>
- Cámara Colombiana de Comercio Electrónico. (2024). *3 tendencias de pagos que impulsan el éxito en el eCommerce*. CCCE. Recuperado de <https://ccce.org.co/noticias/3-tendencias-de-pagos-que-impulsan-el-exito-en-el-ecommerce/>

- Comisión Económica para América Latina y el Caribe (CEPAL). (2021). *El trabajo intermediado por plataformas en Colombia*. Recuperado de <https://repositorio.cepal.org/handle/11362/47011>
- Comisión Económica para América Latina y el Caribe (CEPAL) & Comisión Europea. (2013). *Economía digital y del cuidado*. Recuperado de <https://www.ccce.org.co/noticias/economia-digital-y-del-cuidado/>
- Congreso de Colombia. (2012). *Ley 1581 de 2012 - Por la cual se dictan disposiciones generales para la protección de datos personales*. Recuperado de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- Congreso de Colombia. (2014). *Ley 1735 de 2014 - Por la cual se dictan medidas tendientes a promover el acceso a los servicios financieros transaccionales y se dictan otras disposiciones*. Recuperado de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=59835>
- Congreso de la República de Colombia. (1999). *Ley 527 de 1999: Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones*. Diario Oficial No. 43.673.
- DANE. (2024). *Proyecciones de población municipales a 2025*. Departamento Administrativo Nacional de Estadística. Recuperado de <https://www.dane.gov.co/>
- Departamento Nacional de Planeación. (2019). *Política nacional para la transformación digital e inteligencia artificial (Documento CONPES 3975)*. Departamento Nacional de Planeación. Recuperado de <https://colaboracion.dnp.gov.co/CDT/Conpes/Econ%C3%B3micos/3975.pdf>
- Dumar Hoyos, J. A., & Domínguez Aguirre, J. F. (2023). *Transformación digital rural: impulsando el desarrollo tecnológico y la inclusión en comunidades rurales* [Tesis de maestría, Universidad Cooperativa de Colombia]. Repositorio Institucional Universidad Cooperativa de Colombia. <https://hdl.handle.net/20.500.12494/56762>
- El País. (2024, octubre 17). *La Cámara de Representantes aprueba la reforma laboral de Petro, pero elimina el contrato agropecuario que buscaba formalizar a millones de campesinos*. Recuperado de <https://elpais.com/america-colombia/2024-10-17/la-camara-de-representantes-aprueba-la-reforma-laboral-de-petro-pero-elimina-el-contrato-agropecuario-que-buscaba-formalizar-a-millones-de-campesinos.html>
- El País. (2025, enero 22). *Bancolombia y Davivienda: ¿Cómo hacer transferencias bancarias digitales gratis en Colombia?* Recuperado de <https://elpais.com/america->

colombia/2025-01-22/bancolombia-y-da-vivienda-como-hacer-transferencias-bancarias-digitales-gratis-en-colombia.html

Elluminati Inc. (2023). *Desarrollo de aplicaciones de entrega de alimentos*. Recuperado de <https://www.elluminatiinc.com/es/desarrollo-aplicaciones-entrega-alimentos/>

Fedesarrollo. (2021). *Las plataformas digitales, la productividad y el empleo en Colombia*. Recuperado de <https://www.repository.fedesarrollo.org.co/handle/11445/3962>

Firebase. (2025). *Cloud Firestore*. Recuperado de <https://firebase.google.com/products/firestore>

Firebase. (2025). *Realtime Database*. Recuperado de <https://firebase.google.com/products/realtime-database>

Flutter. (2025). *Build apps for any screen*. Recuperado de <https://flutter.dev/>

Flutter. (2025). *Flutter SDK overview*. Recuperado de <https://docs.flutter.dev/tools/sdk>

Flutter. (2025). *Intro to Dart*. Recuperado de <https://docs.flutter.dev/get-started/fundamentals/dart>

Food App. (2024). *5 características clave para una aplicación móvil de entrega de comida*. Recuperado de <https://www.food-app.io/5-caracteristicas-clave-para-una-aplicacion-movil-de-entrega-de-comida/>

Forbes Staff. (2024, abril 3). *En Colombia hay más de 1.2 millones de conductores y más de 645.000 repartidores en las apps, según Alianza In*. *Forbes Colombia*. Recuperado de <https://forbes.co/2024/04/03/tecnologia/en-colombia-hay-mas-de-1-2-millones-de-conductores-y-mas-de-645-000-repartidores-en-las-apps-segun-alianza-in/>

Fortinet. (2025). *¿Qué es la seguridad de aplicaciones móviles?* Recuperado de <https://www.fortinet.com/lat/resources/cyberglossary/mobile-app-security>

Gironés, J. T. (2013). *El Gran Libro de Android*. Marcombo.

Global Trade Culture. (2024). *Evolución del comercio digital e impacto en economías*. Recuperado de <https://globaltradeculture.com/evolucion-del-comercio-digital-e-impacto-en-economias/>

Gobernación de Caldas. (2025). *Información municipal de Palestina*. Recuperado de <https://site.caldas.gov.co>

- González, L. (2023). *Plataformas digitales y su cobertura territorial en Colombia*. Observatorio TIC Rural – Ministerio TIC. Recuperado de <https://www.mintic.gov.co/>
- Hedyla. (2023). *¿Cómo se aplica la distribución de última milla en zonas rurales?* Recuperado de <https://hedyla.com/retos-de-la-ultima-milla-en-zonas-rurales/>
- Kantar. (2020). *Estudio sobre el uso de aplicaciones móviles para pedidos de domicilios en Colombia*. Kantar.
- Maldonado, O. J., & Vargas, D. S. (2024). *The challenges for fair work in digital platforms in Colombia: A critical exploration on embodied justice and responsibility*. En K. Ogunyemi (Ed.), *Humanistic Management in the Global South*. ResearchGate. Recuperado de <https://www.researchgate.net/publication/383087370>
- McKinsey & Company. (2024). *Pedir a domicilio: La rápida evolución del reparto de comida*. Recuperado de <https://www.mckinsey.com/featured-insights/destacados/pedir-a-domicilio-la-rapida-evolucion-del-reparto-de-comida/es>
- Microsoft. (2023). *Características de una buena aplicación móvil*. Recuperado de <https://www.microsoft.com/en-us/power-platform/products/power-apps/topics/app-development/what-makes-a-good-app>
- Ministerio de Hacienda y Crédito Público. (2022). *Decreto 1297 de 2022 - Por medio del cual se modifica el Decreto 2555 de 2010 en lo relacionado con la regulación de las finanzas abiertas en Colombia y se dictan otras disposiciones*. Recuperado de <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=125957>
- Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC). (2021). *Estudio sobre logística de última milla*. Observatorio de eCommerce. Recuperado de https://observatorioecommerce.mintic.gov.co/797/articles-198595_recurso_1.pdf
- Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC). (2021). *Plan Nacional de Tecnología de la Información y las Comunicaciones (PLAN TIC)*. Recuperado de <https://revistas.upel.edu.ve/index.php/dialectica/article/download/2306/2390/5282>
- Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC). (2025). *Política de Seguridad Digital*. Recuperado de <https://mintic.gov.co/portal/715/w3-article-15430.html>
- Naciones Unidas, Konrad-Adenauer-Stiftung e.V., & Banco Interamericano de Desarrollo. (2021). *Recuperación económica tras la pandemia de COVID-19*. Comisión

Económica para América Latina y el Caribe.

https://repositorio.cepal.org/bitstream/handle/11362/47308/1/S2100678_es.pdf

Observatorio de Comercio Electrónico. (2021). *Análisis normativo del comercio electrónico en Colombia*. Recuperado de

https://observatorioecommerce.mintic.gov.co/797/articles-198739_recurso_1.pdf

OCDE. (2011). *OECD guide to measuring the information society*. OECD Publishing. Recuperado el 10 de julio de 2021, de https://read.oecd-ilibrary.org/science-and-technology/oecd-guide-to-measuring-the-information-society-2011_9789264113541-en#page3

OpenAI. (2026). *ChatGPT (versión GPT-4o) [modelo de lenguaje]*.

<https://chat.openai.com/>

Pay U, Evertec / Place to Pay, PayValida, ePayco, Mercado Pago, & Wompi. (2024).

Categorías de ventas de comercio electrónico y participación de medios de pago.

Payments and Commerce Market Intelligence. (2024). *Datos de mercado: Comercio electrónico en Colombia*. Payments and Commerce Market Intelligence.

<https://paymentscmi.com/insights/datos-mercado-comercio-electronico-colombia/>

PedidosYa. (2024). *Sitio web oficial de PedidosYa*. Recuperado el 1 de junio de 2025, de

<https://www.pedidosya.com/>

Presidencia de la República de Colombia. (2022). *Decreto 338 de 2022 - Por el cual se establece el Modelo de Gobernanza de Seguridad Digital*. Recuperado de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=181866>

Proships. (2023). *Los desafíos y oportunidades de la logística de última milla en áreas rurales o remotas*. Recuperado de <https://proships.com/los-desafios-y-oportunidades-de-la-logistica-de-ultima-milla-en-areas-rurales-o-remotas/>

Quickworks. (2024). *12 Key Features to Look for in a Ready-Made Food Delivery App Solution*. Recuperado de <https://quickworks.medium.com/12-key-features-to-look-for-in-a-ready-made-food-delivery-app-solution-aedc8106a5a7>

Ramírez, L., & Sepúlveda, J. (2018). *Perspectiva rural sobre inclusión digital: un acercamiento a la apropiación de las TIC en comunidades rurales colombianas*

[Tesis de maestría, Corporación Universitaria Minuto de Dios]. Repositorio Institucional UNIMINUTO. <https://repositorio.uniminuto.edu/bitstreams/037bc9e9-1293-4534-864c-bb70d455b107/download>

- Rappi. (2024). *Sitio web oficial de Rappi*. Recuperado el 1 de junio de 2025, de <https://www.rappi.com/>
- Rappi. (2025). *Nuestra historia*. Recuperado de <https://about.rappi.com/es/nuestra-historia>
- Redwerk. (2023). *Cómo crear una aplicación de entrega de comida*. Recuperado de <https://redwerk.es/blog/como-crear-una-aplicacion-de-entrega-de-comida/>
- Redwerk. (2024). *Mejores herramientas de desarrollo móvil multiplataforma 2025*. Recuperado de <https://redwerk.es/blog/mejores-herramientas-de-desarrollo-movil-multiplataforma/>
- Revista La Barra. (2023). *Cómo se está moviendo el negocio de los domicilios en Colombia*. Recuperado de <https://www.revistalabarra.com/es/noticias/como-se-esta-moviendo-el-negocio-de-los-domicilios-en-colombia>
- Sistema Económico Latinoamericano y del Caribe (SELA). (2023). *La economía digital puede generar impacto de USD114 mil millones en Colombia*. Recuperado de <https://www.sela.org/la-economia-digital-puede-generar-impacto-de-usd114-mil-millones-en-colombia/>
- Statista. (2024). *El mercado online mundial de reparto de comida (Food Delivery)*. Recuperado de <https://es.statista.com/temas/8657/el-mercado-online-mundial-de-reparto-de-comida-food-delivery/>
- Statista. (2024). *Valor de mercado de Rappi*. Recuperado de <https://es.statista.com/estadisticas/1253569/valor-de-mercado-de-rappi/>
- Tobar, E. G. (2015). *Impacto del comercio electrónico en las grandes y medianas empresas de la ciudad de Sogamoso Boyacá, Colombia*. *Cuadernos Latinoamericanos de Administración*, 11(20), 83-96.
- Universidad de América. (2023). *Impacto del uso de tecnologías digitales en los procesos de logística en el sector agropecuario de Colombia*. Recuperado de <https://repository.uamerica.edu.co/bitstreams/7ce581d8-aa56-4155-afd5-680f693abf5c/download>
- Universidad Pontificia Bolivariana. (2020). *Formalización del trabajo en plataformas digitales de domicilio*. Recuperado de <https://repository.upb.edu.co/handle/20.500.11912/6166>