

Taller 01

Presentado por:

Juan José Martín Vargas – jmartinv@unal.edu.co

Juan Camilo Posso Portilla – jpossop@unal.edu.co

Esteban Prieto Lugo - eprietol@unal.edu.co

Juan Camilo Vergara Tao – juvergarat@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez

oalvarezr@unal.edu.co

Mayo 9 de 2025



Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas y Computación

2025-1S

Lista de requerimientos

- **Req_1:** Implementación de un sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas).
- **Req_2:** La aplicación debe ser accesible desde la computadora en la oficina.
- **Req_3:** La aplicación debe contar con una versión móvil sencilla para los repartidores (para dispositivos Android y iOS).
- **Req_4:** Unificar la comunicación con un sistema de mensajería interna entre repartidores y central.
- **Req_5:** Implementación de una integración para recibir pedidos directamente desde WhatsApp.
- **Req_6:** La aplicación debe tener respuestas automáticas para cuando el personal se encuentra fuera de horario.
- **Req_7:** El sistema debe permitir modificar rutas rápidamente, y notificar automáticamente los retrasos.
- **Req_8:** Que los pedidos individuales deben cargarse en menos de un minuto.
- **Req_9:** El sistema debe tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.
- **Req_10:** La central debería ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores.
- **Req_11:** Los clientes deben poder cambiar direcciones o cancelar pedidos en el momento.
- **Req_12:** El sistema debe enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.
- **Req_13:** Que haya actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados.
- **Req_14:** La aplicación no debe tener altos tiempos de espera y carga (máx 30 segundos).
- **Req_15:** Que la aplicación pueda funcionar cuando no hay señal.
- **Req_16:** Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera.
- **Req_17:** El sistema debe tener una interfaz que sea llamativa y amable para los usuarios.

Tipos de requerimientos

FUNCIONALES	NO FUNCIONALES
→ (RF_1) El sistema deberá disponer de un rol agente_central que permita el seguimiento en tiempo real de los repartidores. (disponibilidad, ubicación, entregas) (<i>antiguo Req_1</i>).	→ (RNF_1) El sistema debe ser compatible con múltiples plataformas y debe incluir una aplicación de escritorio que pueda utilizarse desde la central, ya sea en Windows o macOS. (<i>antiguo Req_2</i>).

- **(RF_2)** El sistema deberá integrar un subsistema de mensajería interna que permita la transmisión de mensajes en tiempo real entre los roles de usuario agente_central y repartidor (*antiguo Req_4*).
- **(RF_3)** El sistema deberá incorporar una integración con WhatsApp, permitiendo la recepción y procesamiento automático de pedidos enviados por los clientes, que serán gestionados por usuarios con el rol de agente_central (*antiguo Req_5*).
- **(RF_4)** El sistema deberá contar con un bot automatizado que genere respuestas predefinidas de forma automática cuando se detecte una interacción fuera del horario establecido (*antiguo Req_6*).
- **(RF_5)** El sistema debe ser capaz de modificar rutas rápidamente, y notificar automáticamente los retrasos (*antiguo Req_7*).
- **(RF_6)** El sistema debería permitir a la central acciones como ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores (*antiguo Req_10*).
- **(RF_7)** El sistema debe permitir cambios en las direcciones o cancelar pedidos en el momento. (*antiguo Req_11*).
- **(RF_8)** El sistema debería enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace (*antiguo Req_12*).
- **(RF_9)** El sistema debe realizar actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados para que el agente_central y el repartidor pueda verlas (*antiguo Req_13*).
- **(RF_10)** El sistema podría anticipar problemas de tráfico, clima, entre otros, antes de que estos sucedan y buscar una solución óptima para solventarlos de la mejor manera (*antiguo Req_16*).

- **(RNF_2)** La aplicación deberá incluir una versión móvil ligera y optimizada para dispositivos Android e iOS, diseñada específicamente para el uso eficiente por parte del rol de repartidor (*antiguo Req_3*).
- **(RNF_3)** El sistema deberá permitir que los pedidos individuales deben cargarse en menos de un minuto (*antiguo Req_8*).
- **(RNF_4)** Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras (*antiguo Req_9*).
- **(RNF_5)** La aplicación no debe tener altos tiempos de espera y carga (máx 30 segundos) (*antiguo Req_14*).
- **(RNF_6)** El sistema deberá seguir en funcionamiento así el dispositivo no tenga una conexión a internet (*antiguo Req_15*).
- **(RNF_7)** El sistema debe tener una interfaz que sea llamativa y amable para los usuarios (*antiguo Req_17*).

<p>→ (RF_11) El sistema debe poseer un login que permite seleccionar diferentes roles (central y repartidor) <i>(nuevo Req)</i>.</p> <p>→ (RF_12) El sistema deberá contar con un sign up para la creación y registro de los usuarios de los roles (trabajadores y repartidores) <i>(nuevo Req)</i>.</p> <p>→ (RF_13) Creación de base de datos de trabajadores, repartidores, clientes y pedidos <i>(nuevo Req)</i>.</p> <p>→ (RF_14) Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente_central <i>(nuevo Req)</i>.</p> <p>→ (RF_15) Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes <i>(nuevo Req)</i>.</p>	
---	--

Tabla de MoSCoW

Must	Should
<p>→ (RF_1) El sistema deberá disponer de un rol agente_central que permita el seguimiento en tiempo real de los repartidores.</p> <p>→ (RF_2) El sistema deberá integrar un subsistema de mensajería interna que permita la transmisión de mensajes en tiempo real entre los roles de usuario agente_central y repartidor.</p> <p>→ (RF_3) El sistema deberá incorporar una integración con WhatsApp, permitiendo la recepción y procesamiento automático de pedidos enviados por los clientes, que serán gestionados por usuarios con el rol de agente_central.</p> <p>→ (RF_5) El sistema debe ser capaz de modificar rutas rápidamente, y notificar automáticamente los retrasos.</p>	<p>→ (RF_4) El sistema deberá contar con un bot automatizado que genere respuestas predefinidas de forma automática cuando se detecte una interacción fuera del horario establecido.</p> <p>→ (RF_8) El sistema debería enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.</p> <p>→ (RNF_5) La aplicación no debe tener altos tiempos de espera y carga (máx 30 segundos).</p>

- | | |
|--|--|
| <ul style="list-style-type: none">→ (RF_6) El sistema debería permitir a la central acciones como ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores.→ (RF_7) El sistema debe permitir cambios en las direcciones o cancelar pedidos en el momento.→ (RF_9) El sistema debe realizar actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados por el agente_central para que el repartidor pueda verlas.→ (RF_11) El sistema debe poseer un login que permite seleccionar diferentes roles (central y repartidor).→ (RF_12) El sistema deberá contar con un signup para la creación y registro de los usuarios de los roles (trabajadores y repartidores).→ (RF_13) Creación de base de datos de trabajadores, repartidores, clientes y pedidos.→ (RF_14) Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente_central.→ (RF_15) Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes.→ (RNF_1) El sistema debe ser compatible con múltiples plataformas y debe incluir una aplicación de escritorio que pueda utilizarse desde la central, ya sea en Windows o macOS.→ (RNF_2) La aplicación deberá incluir una versión móvil ligera y optimizada para dispositivos Android e iOS, diseñada específicamente para el uso eficiente por parte del rol de repartidor.→ (RNF_3) El sistema deberá permitir que los pedidos individuales deben cargarse en menos de un minuto. | |
|--|--|

<p>Could</p> <p>→ (RNF_4) Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.</p> <p>→ (RNF_7) El sistema debe tener una interfaz que sea llamativa y amable para los usuarios.</p>	<p>Won't</p> <p>→ (RF_10) El sistema podría anticipar problemas de tráfico, clima, entre otros, antes de que estos sucedan y buscar una solución óptima para solventarlos de la mejor manera.</p> <p>→ (RNF_6) El sistema deberá seguir en funcionamiento así el dispositivo no tenga una conexión a internet</p>
---	--

Priorización por puntos de Fibonacci

Requisito		Estimación	Argumento
RF_1	El sistema deberá disponer de un rol agente_central que permita el seguimiento en tiempo real de los repartidores. (disponibilidad, ubicación, entregas).	13	La implementación de este sistema de seguimiento es más compleja, ya que requiere la constante actualización de la ubicación del pedido y los posibles cambios en rutas.
Must			
RF_2	El sistema deberá integrar un subsistema de mensajería interna que permita la transmisión de mensajes en tiempo real entre los roles de usuario agente_central y repartidor.	13	La implementación de un chat interno entre la central y los repartidores puede demandar diferentes niveles de complejidad y recursos.
Must			
RF_3	El sistema deberá incorporar una integración con WhatsApp, permitiendo la recepción y procesamiento automático de pedidos	5	La implementación de este canal no tendría una amplia dificultad por eso tendría una escala de 5 días para su realización.
Must			

	enviados por los clientes, que serán gestionados por usuarios con el rol de agente_central.		
RF_4	El sistema deberá contar con un bot automatizado que genere respuestas predefinidas de forma automática cuando se detecte una interacción fuera del horario establecido.	2	Hace parte de la implementación de un chatbot con la API de WhatsApp para las respuestas automáticas fuera de horario; el tiempo se justifica por la necesidad de escribir los mensajes y que sean aprobados por el cliente.
Should			
RF_5	El sistema debe ser capaz de modificar rutas rápidamente, y notificar automáticamente los retrasos.	13	El desarrollo de cambios en vivo y notificaciones instantáneas requiere tiempo para investigar cómo se debe implementar y posteriormente testear.
Must			
RF_6	El sistema debería permitir a la central acciones como ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores.	13	Las múltiples funcionalidades de este requisito demandan investigación y tiempo suficiente para hacer una primera entrega de calidad.
Must			
RF_7	El sistema debe permitir cambios en las direcciones o cancelar pedidos en el momento.	8	Para la implementación de esta funcionalidad se debe guardar un registro automático de los cambios y notificar a la central para que se realice el cambio más adecuado en la ruta de los repartidores.
Must			

RF_8	El sistema debería enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.	8	Hace parte de la implementación de un chatbot con la API de WhatsApp para las respuestas automáticas con envío de actualizaciones automáticas
Should			
RF_9	El sistema debe realizar actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados por el agente_central para que el repartidor pueda verlas.	8	Este requisito está relacionado en su desarrollo con otros requisitos y su implementación no implicaría un esfuerzo mayor, estimamos 8 días para su desarrollo.
Must			
RF_10	El sistema podría anticipar problemas de tráfico, clima, entre otros, antes de que estos sucedan y buscar una solución óptima para solventarlos de la mejor manera.	N/A	Para una primera entrega se estima tener un tipo de modelo de IA que pueda estar leyendo datos continuamente y que saque conclusiones de cómo está el clima, tráfico, noticias, ETC. y pueda hacer una estimación de cuando pueda haber un estado de caos. No se estima un tiempo tentativo para el desarrollo de este requisito para esta primera entrega.
Won't			
RF_11	El sistema debe poseer un login que permite seleccionar diferentes roles (central y repartidor).	8	Al ser una primera entrega, se podría desarrollar un login tentativo, estimamos 8 días en hacer la investigación y el desarrollo de un login que permita el ingreso de usuarios dependiendo del rol que desempeñan.
Must			

RF_12	El sistema deberá contar con un signup para la creación y registro de los usuarios de los roles (trabajadores y repartidores).	5	Se podría desarrollar un signup tentativo, estimamos 5 días en hacer la investigación y el desarrollo de un login que permite la creación de usuarios dependiendo del rol que desempeñan. Relación directa con la creación de un login.
Must			
RF_13	Creación de base de datos de los agentes de la central, repartidores y pedidos.	8	Para hacer la correcta implementación y análisis de los distintos procesos almacenados como triggers y otros justifica el tiempo.
Must			
RF_14	Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente_central.	8	La implementación de esta API no requiere mayor dificultad, sin embargo el rastreo de los pedidos desde el rol de agente_central. si puede generar un esfuerzo significativo.
Must			
RF_15	Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes.	5	La implementación de este canal no tendría una amplia dificultad por eso tendría una escala de 5 días para su realización.
Must			
RNF_1	El sistema debe ser compatible con múltiples plataformas y debe incluir una aplicación de escritorio que pueda utilizarse desde la central, ya sea en Windows o macOS.	13	Empezando que se va a crear una app desde cero que contenga todas las funciones necesarias para que los repartidores puedan cumplir con sus exigencias se llega a ese estimado para una primera versión.
Must			
RNF_2	La aplicación deberá incluir una versión móvil ligera y	21	Empezando que se va a crear una app desde cero que contenga todas las funciones

Must	optimizada para dispositivos Android e iOS, diseñada específicamente para el uso eficiente por parte del rol de repartidor.		necesarias para que los repartidores puedan cumplir con sus exigencias se llega a ese estimado para una primera versión.
RNF_3	El sistema deberá permitir que los pedidos individuales deben cargarse en menos de un minuto.	21	Con la implementación de un buen algoritmo y un formato JSON para los pedidos por adelantado se puede crear esta funcionalidad de manera sencilla con un tiempo moderado de desarrollo.
Must			
RNF_4	Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.	13	Para implementar estadísticas generales, de modo que la central pueda establecer medidas de calidad y beneficios para los repartidores más ágiles, se requiere establecer algoritmos y acumuladores en los perfiles de los repartidores de modo que se actualicen automáticamente estos números, lo cuál requiere tiempo de planeación, desarrollo y testeo.
Could			
RNF_5	La aplicación no debe tener altos tiempos de espera y carga (máx 30 segundos).	8	Buscamos iniciar desde los procesos que ya tenemos con la app y buscamos que sean lo más optimizados posibles para que no nos utilice espacio para que podamos ya sea optimizarla o mejorarla sin seguir ampliando su tamaño.
Should			

RNF_6	El sistema deberá seguir en funcionamiento así el dispositivo no tenga una conexión a internet.	N/A	Se buscaría crear una manera de que pueda tener chats con la central, buscar la manera de que el repartidor sepa cuando se cambia las direcciones o pedidos a última y hora Sistema de GPS incorporados sin señal. Para este requisito no se estimó el puntaje de Fibonacci ya que requiere bastante tiempo de implementación y para una primera entrega no se desarrollaría.
Won't			
RNF_7	El sistema debe tener una interfaz que sea llamativa y amable para los usuarios.	13	Para el desarrollo de una interfaz llamativa y amigable, se necesita implementar un framework de JavaScript para el frontend, como react o vue, además de la planeación del diseño (colores, distribución del espacio, estilos, entre otros).
Could			