

# Taller 01

## Presentado por:

Juan José Martín Vargas – [jmartinv@unal.edu.co](mailto:jmartinv@unal.edu.co)

Juan Camilo Posso Portilla – [jpossop@unal.edu.co](mailto:jpossop@unal.edu.co)

Esteban Prieto Lugo - [eprietol@unal.edu.co](mailto:eprietol@unal.edu.co)

Juan Camilo Vergara Tao – [juvergarat@unal.edu.co](mailto:juvergarat@unal.edu.co)

## Profesor:

Oscar Eduardo Alvarez Rodriguez

[oalvarezr@unal.edu.co](mailto: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

- Sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, entre otros).
- Accesible desde la computadora en la oficina.
- Versión móvil sencilla para los repartidores (Android y iOS).
- Unificar la comunicación con un sistema de mensajería interna entre repartidores y central.
- Integración para recibir pedidos directamente desde WhatsApp.
- Tener respuestas automáticas cuando estamos fuera de horario.
- Que el sistema permite modificar rutas rápidamente, y notificar automáticamente los retrasos.
- Que los pedidos puedan cargarse rápido.
- Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.
- La central debería ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores.
- Que los clientes puedan cambiar direcciones o cancelar pedidos en el momento.
- Enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.
- Que haya actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados.
- La app no puede ser pesada ni complicada.
- Que la aplicación pueda funcionar cuando no hay señal.
- Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera.
- Interfaz que sea llamativa y amable para los usuarios.

## Tipos de requerimientos

<b>FUNCIONALES</b>	<b>NO FUNCIONALES</b>
<ul style="list-style-type: none"><li>→ Sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, entre otros).</li><li>→ Unificar la comunicación con un sistema de mensajería interna entre repartidores y central.</li></ul>	<ul style="list-style-type: none"><li>→ Accesible desde la computadora en la oficina.</li><li>→ Versión móvil sencilla para los repartidores (Android y iOS).</li><li>→ Que los pedidos puedan cargarse rápido.</li></ul>

<ul style="list-style-type: none"> <li>→ Integración para recibir pedidos directamente desde WhatsApp.</li> <li>→ Tener respuestas automáticas cuando estamos fuera de horario.</li> <li>→ Que el sistema permite modificar rutas rápidamente, y notificar automáticamente los retrasos.</li> <li>→ La central debería ver pedidos, crear rutas, hacer cambios, ver dónde están los repartidores.</li> <li>→ Que los clientes puedan cambiar direcciones o cancelar pedidos en el momento.</li> <li>→ Enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.</li> <li>→ Que haya actualizaciones en tiempo real de los cambios en las rutas y los pedidos activos o cancelados.</li> <li>→ Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera.</li> </ul>	<ul style="list-style-type: none"> <li>→ Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.</li> <li>→ La app no puede ser pesada ni complicada.</li> <li>→ Que la aplicación pueda funcionar cuando no hay señal.</li> <li>→ Interfaz que sea llamativa y amable para los usuarios.</li> </ul>
--	---

## Tabla de MoSCoW

Must	Should
<ul style="list-style-type: none"> <li>→ Login que permite diferentes roles (central y repartidor).</li> <li>→ El usuario con el rol de agente_central debe poder ver el estado de los pedidos, modificar rutas y tener un chat directo con el rol de repartidor.</li> <li>→ El usuario con el rol de repartidor debe poder ver rutas de entregas, marcar pedidos como completados, notificar accidentes e inconvenientes y</li> </ul>	<ul style="list-style-type: none"> <li>→ Que los clientes puedan cambiar direcciones o cancelar pedidos en el momento.</li> <li>→ Tener respuestas automáticas cuando estamos fuera de horario.</li> <li>→ Enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.</li> </ul>

<p>tener un chat directo con el rol de agente_central.</p> <ul style="list-style-type: none"> <li>→ Creación y registro de los usuarios de los roles (trabajadores y repartidores).</li> <li>→ Creación de base de datos de trabajadores, repartidores, clientes y pedidos.</li> <li>→ Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente_central.</li> <li>→ Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes.</li> <li>→ Unificar la comunicación con un sistema de mensajería interna entre repartidores y central.</li> <li>→ Creación de sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, cambios de rutas).</li> <li>→ Optimización en el proceso de carga de los pedidos, permitiendo pedidos en gran cantidad por adelantado.</li> <li>→ Integración con WhatsApp para carga de pedidos automáticos por chat.</li> <li>→ Que haya actualizaciones en tiempo real de los cambios en las rutas, pedidos cancelados e inconvenientes.</li> <li>→ Versión móvil sencilla para los repartidores (Android y iOS).</li> </ul>	
<p><b>Could</b></p> <ul style="list-style-type: none"> <li>→ Interfaz que sea llamativa y amable para los usuarios.</li> <li>→ La app no puede ser pesada ni complicada.</li> </ul>	<p><b>Won't</b></p> <ul style="list-style-type: none"> <li>→ Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera.</li> </ul>

→ Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.	→ Que la aplicación pueda funcionar cuando no hay señal.
--	--

## Priorización por puntos de Fibonacci

Requisito	Priorización	Estimación	Argumento
Creación de sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, cambios de rutas).	Must	21	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.
El usuario con el rol de agente_central debe poder ver el estado de los pedidos, modificar rutas y tener un chat directo con el rol de repartidor.	Must	13	Las funcionalidades de este requisito demandan investigación y tiempo suficiente para hacer una primera entrega de calidad.
El usuario con el rol de agente_central debe poder ver el estado de los pedidos, modificar rutas y tener un chat directo con el rol de repartidor.	Must	13	Las funcionalidades de este requisito demandan investigación y tiempo suficiente para hacer una primera entrega de calidad.
El usuario con el rol de repartidor debe	Must	13	Las funcionalidades de este requisito demandan investigación y tiempo

poder ver rutas de entregas, marcar pedidos como completados, notificar accidentes e inconvenientes y tener un chat directo con el rol de agente_central.			suficiente para hacer una primera entrega de calidad.
Creación de base de datos de trabajadores, repartidores, clientes y pedidos.	Must	13	Para hacer la correcta implementación y análisis de los distintos procesos almacenados como triggers y otros justifica el tiempo
Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente_central.	Must	13	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.
Unificar la comunicación con un sistema de mensajería interna entre repartidores y central.	Must	13	La implementación de un chat interno entre la central y los repartidores puede demandar diferentes niveles de complejidad y recursos.
Login que permite diferentes roles (central y repartidor).	Must	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.

Creación y registro de los usuarios de los roles (trabajadores y repartidores).	Must	8	Se podría desarrollar un sign up tentativo, estimamos 8 días en hacer la investigación y el desarrollo de un login que permite la creación de usuarios dependiendo del rol que desempeñan.
Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes.	Must	8	La implementación de este canal no tendría una amplia dificultad por eso tendría una escala de 8 días para su realización.
Integración con WhatsApp para carga de pedidos automáticos por chat.	Must	8	Esta integración para la carga de pedidos no significa mayor esfuerzo de desarrollo, conectar con chatbots o IA.
Que haya actualizaciones en tiempo real de los cambios en las rutas, pedidos cancelados e inconvenientes.	Must	8	Implementar un chat interno entre la central y los repartidores demandará diferentes niveles de complejidad y recursos.
Que los clientes puedan cambiar direcciones o cancelar pedidos en el momento.	Should	8	Como hace parte del chat incorporado no conlleva mucha demanda del tiempo
Tener respuestas automáticas cuando estamos fuera de horario.	Should	3	Hace parte de la implementación de un chatbot con la API de WhatsApp para las respuestas automáticas fuera de horario.

Enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace.	Should	3	Hace parte de la implementación de un chatbot con la API de WhatsApp para las respuestas automáticas con envío de actualizaciones automáticas.
Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras.	Could	8	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.
Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera.	Won't	89	Para una primera entrega se estima tener un tipo de modelo de IA que pueda estar leyendo datos continuamente y que saque conclusiones de como está el clima, tráfico, noticias, ETC. y pueda hacer una estimación de cuando pueda haber un estado de caos.
Que la aplicación pueda funcionar cuando no hay señal.	Won't	55	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.
Versión móvil sencilla para los repartidores (Android y iOS).	Must	21	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.



Optimización en el proceso de carga de los pedidos, permitiendo pedidos en gran cantidad por adelantado.	Must	13	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.
Interfaz que sea llamativa y amable para los usuarios.	Could	13	El desarrollo de una interfaz llamativa y amigable, estaríamos implementando paquetes de frontend como bootstrap conllevando de esta manera a que se anexe un tiempo extra.
La app no puede ser pesada ni complicada.	Could	13	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.