**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*

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

| **FUNCIONALES** | **NO FUNCIONALES** |
| --- | --- |
| * Sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, entre otros). * 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. * 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. * Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera. | * Accesible desde la computadora en la oficina. * Versión móvil sencilla para los repartidores (Android y iOS). * Que los pedidos puedan cargarse rápido. * Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras. * La app no puede ser pesada ni complicada. * Que la aplicación pueda funcionar cuando no hay señal. * Interfaz que sea llamativa y amable para los usuarios. |

# 

# Tabla de MoSCoW

| **Must**   * Login que permite diferentes roles (central y repartidor). * 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. * El usuario con el rol de repartidor debe poder ver rutas de entregas, marcar pedidos como completados, notificar accidentes e inconvenientes y tener un chat directo con el rol de agente\_central. * Creación y registro de los usuarios de los roles (trabajadores y repartidores). * Creación de base de datos de trabajadores, repartidores, clientes y pedidos. * Implementación de una API de Google Maps para el rastreo en tiempo real de los pedidos y repartidores desde el rol de agente\_central. * Implementación de WhatsApp que permita la integración de los canales de comunicación entre clientes. * Unificar la comunicación con un sistema de mensajería interna entre repartidores y central. * Creación de sistema de seguimiento en tiempo real de repartidores (disponibilidad, ubicación, entregas, cambios de rutas). * Optimización en el proceso de carga de los pedidos, permitiendo pedidos en gran cantidad por adelantado. * Integración con WhatsApp para carga de pedidos automáticos por chat. * Que haya actualizaciones en tiempo real de los cambios en las rutas, pedidos cancelados e inconvenientes. * Versión móvil sencilla para los repartidores (Android y iOS). | **Should**   * Que los clientes puedan cambiar direcciones o cancelar pedidos en el momento. * Tener respuestas automáticas cuando estamos fuera de horario. * Enviar actualizaciones automáticas a los clientes del estado del pedido o a través de un enlace. |
| --- | --- |
| **Could**   * Interfaz que sea llamativa y amable para los usuarios. * La app no puede ser pesada ni complicada. * Tener estadísticas sobre tiempos de entrega, zonas más problemáticas, repartidores más rápidos, entre otras. | **Won’t**   * Que pueda anticipar problemas antes de que sucedan como el tráfico, clima y busque una solución óptima para solventarlos de la mejor manera. * 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 cambias 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 poder ver rutas de entregas, marcar pedidos como completados, notificar accidentes e inconvenientes y tener un chat directo con el rol de agente\_central. | Must | 13 | Las funcionalidades de este requisito demandan investigación y tiempo 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 signup 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. |