

Estudiantes

- Andrés Felipe Alarcón Pulido - analarconp@unal.edu.co
- Juan Daniel Jossa Soliz - jjossa@unal.edu.co
- Jaime Darley Angulo Tenorio - janqulot@unal.edu.co
- Michel Mauricio Castañeda Braga - micastanedab@unal.edu.co

1. Formar los grupos (el trabajo es grupal...)
 2. Leer nuevamente el chat generado en la práctica de levantamiento de requerimientos (Archivo distinto para cada GP)
 3. Generar una lista de todos los requerimientos identificados dados por el usuario.
- **req 1:** La app debe permitir registrar pedidos fácilmente, ya sea manualmente (con un formulario) o importando desde archivos como Excel. Cada pedido debe incluir datos clave: dirección, cliente, horario y tipo de paquete..
 - **req 2:** El sistema debe asignar pedidos a repartidores de forma clara y permitir crear rutas simples. Las asignaciones podrán ajustarse en tiempo real si hay cambios.
 - **req 3:** La central debe poder monitorear la ubicación de los repartidores y el estado de los pedidos para actuar rápido ante problemas.
 - **req 4:** Los repartidores tendrán una app móvil para ver sus entregas diarias, marcar completadas, recibir actualizaciones y (en el futuro) consultar un mapa de ruta.
 - **req 5:** El sistema debe notificar a los clientes sobre el estado de sus pedidos (mensajes automáticos o enlaces de rastreo). En el futuro, los clientes podrán cargar pedidos directamente.
 - **req 6:** Integrar WhatsApp para recibir pedidos, enviar notificaciones automáticas y responder fuera de horario con mensajes predefinidos.



- **req 7:** Perfiles de usuario diferenciados:
 - o Operadoras: Acceso completo.
 - o Repartidores: Solo funciones de entrega.
 - o Clientes: (Opcional) Seguimiento básico.
- **req 8:** En futuras etapas, el sistema generará reportes de métricas (tiempos de entrega, desempeño de repartidores, zonas problemáticas) para mejorar el servicio.
- **req 9:** La app debe funcionar en computadoras y móviles (Android/iOS), incluso en dispositivos de gama media/baja.
- **req 10:** Interfaz intuitiva en web y móvil, pensada para usuarios con poca experiencia tecnológica (ej. repartidores).
- **req 11:** La app debe ser liviana, rápida y no consumir muchos recursos para un uso fluido durante la jornada..
- **req 12:** La app móvil debe funcionar sin conexión en modo básico y sincronizar datos al recuperar señal.
- **req 13:** Arquitectura escalable para agregar nuevas funciones, usuarios o zonas sin rediseñar el sistema.
- **req 14:** Desarrollo modular y documentado para facilitar futuras actualizaciones o correcciones.
- **req_15:** Seguridad de datos con roles de acceso y protocolos para manejar información sensible.
- **req 16:** El sistema debe estar activo sin fallas en horario laboral y responder automáticamente por WhatsApp fuera de él.

4. Dividir los requerimientos en funcionales o no funcionales

FUNCIONALES	NO FUNCIONALES
<ul style="list-style-type: none">- RF_1: La aplicación debe permitir a las operadoras registrar nuevos pedidos desde la oficina de forma ágil, ya sea manualmente a través de un formulario o mediante la importación desde archivos como	<ul style="list-style-type: none">- RNF_1: La solución deberá ser accesible tanto desde computadores de escritorio (para uso en oficina) como desde dispositivos móviles con sistemas Android e iOS, considerando que algunos teléfonos son de gama

<p>hojas de cálculo. Cada pedido deberá contener información clave como dirección, cliente, horario y tipo de paquete.</p> <ul style="list-style-type: none"> - RF_2: El sistema deberá facilitar la asignación de los pedidos a los repartidores de manera clara y organizada, permitiendo además la creación de rutas simples. Será posible modificar estas asignaciones en tiempo real en caso de imprevistos o cambios de última hora. - RF_3: Desde la central de operaciones, se deberá poder monitorear la ubicación actual de los repartidores y el estado de cada pedido. Esto permitirá tener una visión general de la operación y actuar rápidamente ante cualquier incidente. - RF_4: Los repartidores contarán con una versión móvil de la aplicación en la que podrán visualizar su lista diaria de entregas, marcar aquellas ya completadas, recibir actualizaciones sobre cambios en sus rutas y, en etapas futuras, consultar un mapa de su recorrido. - RF_5: La solución deberá incluir mecanismos para mantener informados a los clientes sobre el estado de sus entregas, ya sea mediante mensajes automáticos o enlaces de rastreo. Además, se espera una futura funcionalidad para que los clientes puedan cargar pedidos directamente. - RF_6: Se requiere integrar la aplicación con WhatsApp para facilitar la recepción de pedidos, el envío automático de notificaciones y la atención fuera de horario con respuestas predefinidas. Esta integración es clave para mantener la continuidad de la comunicación con 	<p>media o baja.</p> <ul style="list-style-type: none"> - RNF_2: Tanto la versión web como la app móvil deben tener una interfaz intuitiva, especialmente diseñada para usuarios con bajos niveles de experiencia tecnológica, como los repartidores. - RNF_3: La aplicación debe ser liviana y garantizar tiempos de respuesta rápidos, sin exigir muchos recursos del dispositivo, para asegurar un funcionamiento fluido durante la jornada laboral. - RNF_4: Se deberá contemplar que la aplicación móvil permita operar con funciones básicas incluso sin conexión a internet, y que los datos se sincronicen automáticamente una vez se restablezca la señal. - RNF_5: La arquitectura de la solución deberá permitir incorporar nuevas funcionalidades, usuarios y zonas de operación sin necesidad de rediseñar todo el sistema, acompañando el crecimiento de la empresa. - RNF_6: El software debe estar desarrollado de forma modular y documentada, de modo que futuras mejoras, correcciones o expansiones puedan realizarse de forma eficiente y ordenada. - RNF_7: Es indispensable garantizar que la información esté protegida mediante roles de acceso diferenciados y protocolos adecuados para el manejo de datos sensibles de clientes y entregas. - RNF_8: La solución debe estar disponible durante el horario laboral sin interrupciones, y ofrecer mecanismos automáticos para responder fuera de horario a través
--	---

<p>los clientes en su canal habitual.</p> <ul style="list-style-type: none">- RF_7: El sistema deberá contemplar distintos perfiles de usuario: operadoras con acceso completo al sistema, repartidores con funciones limitadas al cumplimiento de sus entregas, y clientes que podrán acceder (opcionalmente) a información básica de seguimiento.- RF_8: En fases posteriores, el sistema deberá ofrecer reportes sobre métricas clave como tiempos de entrega, desempeño de los repartidores y zonas con más incidencias, con el objetivo de facilitar la mejora continua del servicio.	<p>de WhatsApp, manteniendo la atención al cliente activa en todo momento.</p>
---	--

5. Organizar por MoSCoW

MUST (DEBE - OBLIGATORIO SIN ÉL EL SISTEMA PIERDE SENTIDO O NO FUNCIONA)

- **RF_1:** El usuario con el rol de operadora debe poder registrar nuevos pedidos desde la oficina de forma ágil, ya sea manualmente a través de un formulario o mediante la importación desde archivos como hojas de cálculo.
- **RF_2:** El usuario con el rol de operadora debe poder asignar pedidos a los repartidores y crear rutas simples, con la posibilidad de modificar estas asignaciones en tiempo real.
- **RF_3:** El usuario con el rol de operadora debe poder monitorear la ubicación actual de los repartidores y el estado de cada pedido en tiempo real desde la central de operaciones.
- **RF_4:** El usuario con el rol de repartidor debe poder visualizar su lista diaria de entregas, marcar pedidos como completados y recibir actualizaciones de ruta mediante la aplicación móvil.
- **RF_7:** El sistema debe gestionar distintos perfiles de usuario, otorgando a cada rol (operadora, repartidor, cliente) los accesos y funcionalidades correspondientes.



- **RNF_1:** El sistema debe ser accesible tanto desde computadores de escritorio como desde dispositivos móviles Android e iOS, considerando limitaciones de hardware.
- **RNF_2:** El sistema debe tener una interfaz intuitiva y fácil de usar, adaptada especialmente a usuarios con baja alfabetización digital.
- **RNF_3:** El sistema debe garantizar un rendimiento fluido y tiempos de respuesta rápidos en dispositivos de gama media o baja.
- **RNF_7:** El sistema debe garantizar la seguridad de la información mediante control de accesos por rol y protección de datos sensibles.

SHOULD (DEBERÍA - NO OBLIGATORIO PERO APORTA MUCHO A LA EXP DE USUARIO)

- **RF_5:** El sistema debería permitir informar automáticamente a los clientes sobre el estado de sus entregas, mediante mensajes o enlaces de rastreo, y contemplar en el futuro la posibilidad de que carguen pedidos.
- **RF_6:** El sistema debería integrar funcionalidades con WhatsApp para recibir pedidos, enviar notificaciones automáticas y brindar respuestas fuera de horario.
- **RNF_4:** El sistema debería permitir el funcionamiento básico de la aplicación móvil sin conexión a internet, sincronizando datos al reconectarse.
- **RNF_5:** El sistema debería estar diseñado para ser escalable, permitiendo añadir usuarios, zonas o funciones sin rediseños mayores.
-
- **RNF_6:** El sistema debería ser mantenible, con una arquitectura modular y documentación clara que facilite futuras mejoras o correcciones.
-
- **RNF_8:** El sistema debería garantizar alta disponibilidad durante el horario laboral, e incluir respuestas automáticas fuera de horario para mantener la atención al cliente.

COULD (PODRÍA - ES UN PLUS USUALMENTE PERSONALIZACIÓN, SE DESARROLLA USUALMENTE CUANDO QUEDA TIEMPO)

- **RF_8:** El sistema podría ofrecer reportes y estadísticas sobre tiempos de entrega, desempeño de repartidores y zonas críticas, con el fin de apoyar procesos de mejora continua.

WONT (NO NECESARIO - PUEDE ESPERAR AL PUNTO QUE ESTÉ O NO NO PASA NADA SI NUNCA SE DESARROLLA)

- Todos los requerimientos mencionados aportan valor directo o indirecto al sistema.

6. Realizar una estimación de tiempo/esfuerzo con Fibonacci sobre cada requisito de cuánto se demoraría

Requisito		Estimación	Argumento
RF_1	Registro e importación de pedidos	8	Formularios e importación desde Excel requiere validaciones y lógica específica.
MUST			
RF_2	Asignación de rutas a repartidores	13	Incluye UI para rutas, edición en tiempo real y lógica de asignación dinámica.
MUST			
RF_3	Seguimiento de entregas en tiempo real	13	Requiere geolocalización, comunicación constante y visualización dinámica.
MUST			
RF_4	App móvil para repartidores	13	Requiere diseño móvil, sincronización, notificaciones y posible mapa.
MUST			
RF_5	Notificación y rastreo para clientes	8	Implica mensajes automáticos, links de seguimiento y lógica condicional.
SHOULD			
RF_6	Integración con WhatsApp	13	Integración externa compleja, autenticación y automatización de mensajes.
SHOULD			
RF_7	Gestión de roles y accesos	3	Gestión estándar de roles, bajo impacto técnico si se usa framework.
MUST			
RF_8	Generación de reportes y métricas	8	Requiere acumulación de datos, filtros y generación de estadísticas.
COULD			

RNF_1	Acceso desde PC y móviles	8	Necesita desarrollo multiplataforma y pruebas en distintos dispositivos.
MUST			
RNF_2	Interfaz intuitiva y simple	5	Diseño UI/UX básico, importante pero poco costoso técnicamente.
MUST			
RNF_3	Buen rendimiento en celulares	5	Se resuelve con buenas prácticas de desarrollo.
MUST			
RNF_4	Soporte sin conexión (offline)	8	Manejo de almacenamiento local, sincronización y resolución de conflictos.
SHOULD			
RNF_5	Escalabilidad modular	5	Arquitectura pensada para crecimiento, implica diseño anticipado.
SHOULD			
RNF_6	Mantenibilidad del sistema	5	Documentación y estructura limpia, esfuerzo bajo si se hace desde el inicio.
SHOULD			
RNF_7	Seguridad y protección de datos	8	Necesita control de accesos y medidas de protección razonables.
MUST			
RNF_8	Alta disponibilidad y respuestas fuera de horario	8	Requiere monitoreo, manejo de errores y automatización en canal externo.
SHOULD			