



Estudiantes:

Daniela Ariadna Rueda Hernande Jared Mijail Ramirez Escalante

Silvana Suarez Carvajal Nicolas Rodriguez Piraban

Requerimientos del cliente EcoRuta:

- RQ01. El sistema deberá soportar 3 roles de usuario: “Repartidor”, “Cliente” y “Operador”, cada uno con permisos y funcionalidades específicas.
- RQ02. El sistema debe contar con una interfaz responsive, web y móvil, que permitan su visualización en diferentes dispositivos y plataformas.
- RQ03. El sistema en su aplicación móvil deberá ser compatible tanto con dispositivos iOS como Android
- RQ04. El sistema deberá contar con una aplicación móvil para los roles “Repartidor” y “Cliente”
- RQ06. El usuario de rol “Operador” deberá tener acceso a una interfaz para registrar los pedidos de forma ordenada y estandarizada, así evitar duplicados.
- RQ07. El usuario de rol “Repartidor” deberá poder actualizar el estado del pedido mediante la interfaz móvil, con las siguientes opciones: “En camino al pedido”, “Pedido recogido”, “Pedido en camino” y “Pedido entregado”, asociadas al estado de avance en el proceso de envío.
- RQ08. El usuario de rol “Operador” deberá tener acceso a una interfaz para realizar seguimiento en tiempo real del estado de avance en el proceso de envío del rol “Repartidor”.
- RQ09. El usuario de rol “Cliente” deberá poder consultar el estado de su pedido y visualizar su ubicación en el mapa en tiempo real.
- RQ10. El sistema deberá registrar en tiempo real la ubicación de los repartidores en el mapa y su indicador de disponibilidad.
- RQ11. El usuario de rol “Operador” deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su disponibilidad, esta interfaz cuenta con un listado de todos los repartidores.
- RQ11.1. El usuario de rol “Operador” deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su ubicación, esta interfaz cuenta con la ubicación de todos los repartidores.
- RQ12. El rol de usuario “Cliente” deberá poder modificar o cancelar los pedidos realizados antes de su recolección.

- RQ13. El usuario de rol "Cliente" deberá visualizar a través de la interfaz información fundamental del repartidor asignado a su pedido: nombre, número de envíos realizados y calificación promedio.
- RQ14. El sistema deberá permitir la comunicación bidireccional entre "Operadores", "Clientes" y "Repartidores", vía mensajería integrada.
- RQ15. El sistema debe actualizar dinámicamente las rutas asignadas para optimizar tiempo y reaccionar ante cambios o eventualidades.
- RQ16. El usuario de rol "Operador" deberá notificar al usuario de rol "Repartidor" sobre novedades en el pedido, tales como cambio de dirección o cancelación.
- RQ17. El sistema deberá generar dashboards con estadísticas clave como: tiempos promedio de entrega, zonas con retrasos frecuentes, desempeño de repartidores.
- RQ18 El sistema deberá contar con un ingreso autorizado de forma segura, verificando los datos del usuario y su rol.
- RQ19. El sistema deberá ser resistente a ataques y otras vulnerabilidades de ciberseguridad comunes
- RQ20. El sistema deberá integrar una pasarela de pagos que permita a los usuarios con el rol "Cliente" efectuar el pago correspondiente a sus pedidos de forma segura.
- RQ21. El sistema deberá integrarse con la API de WhatsApp Business para permitir a los "Clientes" realizar pedidos y recibir notificaciones de estado por este canal.
- RQ22. Las notificaciones de cancelación o modificación para los usuarios de tipo "Repartidor" no podrán exceder 5 segundos de delay en su envío.
- RQ23. El sistema en su aplicación móvil debe de poder funcionar en dispositivos de baja gama.
- RQ24. El sistema deberá poder mantener un mínimo 30 conexiones simultáneas con alta estabilidad y rendimiento

Reunión de Requisitos:

Funcionales	No funcionales
RF_1 (RQ01): El sistema debe contar con 3 tipos de clasificación de sus usuarios (roles de usuarios), estos siendo repartidor, Cliente y operador.	RNF_1 (RQ02): El sistema debe contar con una interfaz responsive, web y móvil, que permitan su visualización en diferentes dispositivos y plataformas.
RF_2 (RQ06): El usuario de rol "Operador" deberá tener acceso a una interfaz para registrar los pedidos de forma ordenada y estandarizada, así evitar duplicados.	RNF_2 (RQ03): El sistema en su aplicación móvil deberá ser compatible tanto con dispositivos iOS como Android.
RF_3 (RQ07): El usuario de rol "Repartidor" deberá poder actualizar el estado del pedido mediante la interfaz móvil, con las siguientes opciones: "En camino al pedido", "Pedido recogido", "Pedido en camino" y "Pedido entregado", asociadas al estado de avance en el proceso de envío.	RNF_3 (RQ18): El sistema deberá contar con un ingreso autorizado de forma segura, verificando los datos del usuario y su rol.
RF_4 (RQ08): El usuario de rol "Operador" deberá tener acceso a una interfaz para realizar seguimiento en tiempo real del estado de avance en el proceso de envío del rol "Repartidor".	RNF_4 (RQ19): El sistema deberá ser resistente a ataques y otras vulnerabilidades de ciberseguridad comunes
RF_5 (RQ09): El usuario de rol "Cliente" deberá poder consultar el estado de su pedido y visualizar su ubicación en el mapa en tiempo real.	RNF_5 (RQ21): El sistema deberá integrarse con la API de WhatsApp Business para permitir a los "Clientes" realizar pedidos y recibir notificaciones de estado por este canal.
RF_6 (RQ10): El sistema deberá registrar en tiempo real la ubicación de los repartidores en el mapa y su indicador de disponibilidad.	RNF_6 (RQ22): Las notificaciones de cancelación o modificación para los usuarios de tipo "Repartidor" no podrán exceder 5 segundos de delay en su envío.
RF_7 (RQ11): El usuario de rol "Operador" deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su disponibilidad, esta interfaz cuenta con un listado de todos los repartidores.	RNF_7 (RQ23): El sistema en su aplicación móvil debe de poder funcionar en dispositivos de baja gama.
RF_8 (RQ12): El rol de usuario "Cliente" deberá poder modificar o cancelar los pedidos realizados antes de su recolección.	RNF_8 (RQ24): El sistema deberá poder mantener un mínimo 30 conexiones simultáneas con alta estabilidad y rendimiento

RF_9 (RQ13): El usuario de rol "Cliente" deberá visualizar a través de la interfaz información fundamental del repartidor asignado a su pedido: nombre, número de envíos realizados y calificación promedio.	
RF_10 (RQ14): El sistema deberá permitir la comunicación bidireccional entre dos usuarios del sistema (con diferente rol), vía mensajería integrada.	
RF_11 (RQ15): El sistema debe actualizar dinámicamente las rutas asignadas para optimizar tiempo y reaccionar ante cambios o eventualidades.	
RF_12 (RQ16): El usuario de rol "Operador" deberá notificar al usuario de rol "Repartidor" sobre novedades en el pedido, tales como cambio de dirección o cancelación.	
RF_13 (RQ17): El sistema deberá generar dashboards con estadísticas clave como: tiempos promedio de entrega, zonas con retrasos frecuentes, desempeño de repartidores.	
RF_14 (RQ20): El sistema deberá integrar una pasarela de pagos que permita a los usuarios con el rol "Cliente" efectuar el pago correspondiente a sus pedidos de forma segura.	
RF_15 (RQ11.1): El usuario de rol "Operador" deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su ubicación, esta interfaz cuenta con la ubicación de todos los repartidores.	
RF_16 (RQ04). El sistema deberá contar con una aplicación móvil para los roles "Repartidor" y "Cliente"	

Tabla MoSCoW

Must	Should	Could	Won't
RNF_1 (RQ02): El sistema debe contar con una interfaz responsive, web y móvil, que permitan su visualización en diferentes dispositivos y plataformas.	RF_4 (RQ08): El usuario de rol "Operador" deberá tener acceso a una interfaz para realizar seguimiento en tiempo real del estado de avance en el proceso de envío del rol "Repartidor".	RF_3 (RQ07): El usuario de rol "Repartidor" deberá poder actualizar el estado del pedido mediante la interfaz móvil, con las siguientes opciones: "En camino al pedido", "Pedido recogido", "Pedido en camino" y "Pedido entregado", asociadas al estado de avance en el proceso de envío.	RF_13 (RQ17): El sistema deberá generar dashboards con estadísticas clave como: tiempos promedio de entrega, zonas con retrasos frecuentes, desempeño de repartidores.
RNF_2 (RQ03): El sistema en su aplicación móvil deberá ser compatible tanto con dispositivos iOS como Android.	RF_5 (RQ09): El usuario de rol "Cliente" deberá poder consultar el estado de su pedido y visualizar su ubicación en el mapa en tiempo real.	RF_6 (RQ10): El sistema deberá registrar en tiempo real la ubicación de los repartidores en el mapa y su indicador de disponibilidad.	
RNF_3 (RQ18): El sistema deberá contar con un ingreso autorizado de forma segura, verificando los datos del usuario y su rol.	RF_8 (RQ12): El rol de usuario "Cliente" deberá poder modificar o cancelar los pedidos realizados antes de su recolección.	RF_9 (RQ13): El usuario de rol "Cliente" deberá visualizar a través de la interfaz información fundamental del repartidor asignado a su pedido: nombre, número de envíos realizados y calificación promedio.	
RNF_4 (RQ19): El sistema deberá ser resistente a ataques y otras vulnerabilidades de	RF_10 (RQ14): El sistema deberá permitir la comunicación bidireccional entre	RNF_5 (RQ21): El sistema deberá integrarse con la API de WhatsApp Business para	

ciberseguridad comunes	dos usuarios del sistema (con diferente rol), vía mensajería integrada.	permitir a los "Clientes" realizar pedidos y recibir notificaciones de estado por este canal.	
RF_1 (RQ01): El sistema debe contar con 3 tipos de clasificación de sus usuarios (roles de usuarios), estos siendo repartidor, Cliente y operador.	RF_15 (RQ11.1): El usuario de rol "Operador" deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su ubicación, esta interfaz cuenta con la ubicación de todos los repartidores.	RF_11 (RQ15): El sistema debe actualizar dinámicamente las rutas asignadas para optimizar tiempo y reaccionar ante cambios o eventualidades.	
RF_2 (RQ06): El usuario de rol "Operador" deberá tener acceso a una interfaz para registrar los pedidos de forma ordenada y estandarizada, así evitar duplicados.	RNF 7 (RQ23): El sistema en su aplicación móvil debe de poder funcionar en dispositivos de baja gama.	RNF_6 (RQ22): Las notificaciones de cancelación o modificación para los usuarios de tipo "Repartidor" no podrán exceder 5 segundos de delay en su envío.	
RF_7 (RQ11): El usuario de rol "Operador" deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su disponibilidad, esta interfaz cuenta con un listado de todos los repartidores.	RF_12 (RQ16): El usuario de rol "Operador" deberá notificar al usuario de rol "Repartidor" sobre novedades en el pedido, tales como cambio de dirección o cancelación.		
RF_14 (RQ20): El sistema deberá integrar una pasarela de pagos que permita a los usuarios con el rol "Cliente" efectuar el pago correspondiente a sus pedidos de			

forma segura.			
RNF 8 (RQ24): El sistema deberá poder mantener un mínimo 30 conexiones simultáneas con alta estabilidad y rendimiento			
RF_16 (RQ04). El sistema deberá contar con una aplicación móvil para los roles "Repartidor" y "Cliente"			

Nota: Decidimos usar puntos de historia con los números de fibonacci para nuestra estimación

Requisito		Estimación	Argumento
RF_1	El sistema debe contar con 3 tipos de clasificación de sus usuarios (roles de usuarios), estos siendo repartidor, Cliente y operador	2	Ya que implica definir y almacenar los roles de repartidor, cliente y operador en la base de datos, configurar la lógica de autorización para cada rol en el backend, con la interfaz respectiva
MUST			
RF_2	El usuario de rol "Operador" deberá tener acceso a una interfaz para registrar los pedidos de forma ordenada y estandarizada, así evitar duplicados.	5	Ya que requiere diseñar e implementar en la interfaz del operador un formulario con campos estandarizados y validaciones para evitar duplicados, desarrollar la lógica en el backend que detecte y prevenga registros repetidos al persistir los pedidos, garantizar el control de permisos específico para el rol "Operador" y ejecutar pruebas unitarias
MUST			
RF_3	El usuario de rol "Repartidor" deberá poder actualizar el estado del pedido mediante la interfaz móvil, con las siguientes opciones: "En camino al	5	Ya que requiere diseñar la interfaz móvil con las cuatro opciones de estado y validación de selección. Implementar la lógica de estados en el backend con un endpoint seguro para el rol "Repartidor" y pruebas
COULD			

Requisito		Estimación	Argumento
	pedido”, “Pedido recogido” , “ Pedido en camino" y “Pedido entregado” , asociadas al estado de avance en el proceso de envío.		asociadas.
RF_4	El usuario de rol “Operador” deberá tener acceso a una interfaz para realizar seguimiento en tiempo real del estado de avance en el proceso de envío del rol “Repartidor”.	5	Porque requiere usar una interfaz que muestre en tiempo real los estados enviados por el repartidor. Implementar WebSockets o polling en el backend con autorización para el rol “Operador” y pruebas asociadas.
SHOULD			
RF_5	El usuario de rol “Cliente” deberá poder consultar el estado de su pedido y visualizar su ubicación en el mapa en tiempo real.	8	Porque requiere una interfaz que muestre el estado del pedido y un mapa en tiempo real, integrando un proveedor de mapas y validando permisos del cliente. Implementar un canal de actualización continua (WebSockets o similar) para recibir la ubicación del repartidor y pruebas end-to-end.
SHOULD			
RF_6	El sistema deberá registrar en tiempo real la ubicación de los repartidores en el mapa y su indicador de disponibilidad.	13	Porque requiere procesar la ubicación en tiempo real de lo repartidores y renderizar dinámicamente esta información, lo que también requerirá pruebas
COULD			
RF_7	El usuario de rol “Operador” deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su disponibilidad, esta interfaz cuenta con un listado de todos los repartidores.	5	Porque requiere una interfaz con listado y filtrado de repartidores por disponibilidad y botón de asignación. Lógica backend para validar permisos, actualizar el reparto asignado, gestionar disponibilidad y pruebas.
MUST			
RF_8	El rol de usuario “Cliente” deberá poder modificar o cancelar los pedidos realizados antes de su recolección.	5	Porque requiere crear en la UI opciones para editar o cancelar pedidos antes de la recolección, con validaciones de estado e implementar en el backend la lógica de permisos, actualización o anulación del pedido y sus pruebas asociadas.
SHOULD			

Requisito		Estimación	Argumento
RF_9 COULD	El usuario de rol "Cliente" deberá visualizar a través de la interfaz información fundamental del repartidor asignado a su pedido: nombre, número de envíos realizados y calificación promedio.	3	Porque requiere diseñar en la interfaz la sección que muestre nombre, total de envíos y calificación promedio del repartidor y crear en el backend un endpoint que recupere esos datos agregados con validación de permisos y pruebas.
RF_10 SHOULD	El sistema deberá permitir la comunicación bidireccional entre dos usuarios del sistema (con diferente rol), vía mensajería integrada.	13	Porque requiere diseñar e implementar una interfaz de chat en tiempo real con notificaciones y control de conversaciones entre roles y crear un servicio de mensajería (WebSockets o similar) que almacene mensajes, valide permisos y soporte pruebas end-to-end.
RF_11 COULD	El sistema debe actualizar dinámicamente las rutas asignadas para optimizar tiempo y reaccionar ante cambios o eventualidades.	13	Porque requiere integrar un motor de optimización de rutas que recalcula dinámicamente trayectos ante cambios (tráfico, cancelaciones, nuevas órdenes) y lo expone vía API además de implementar la lógica de programación en tiempo real, manejo de eventos y pruebas de rendimiento para asegurar escalabilidad y fiabilidad.
RF_12 SHOULD	El usuario de rol "Operador" deberá notificar al usuario de rol "Repartidor" sobre novedades en el pedido, tales como cambio de dirección o cancelación.	5	Porque requiere una interfaz para el operador con opciones para enviar notificaciones de cambios o cancelaciones. Un servicio en el backend que envíe estas notificaciones en tiempo real al repartidor, con validación de permisos y pruebas asociadas.
RF_13 WONT	El sistema deberá generar dashboards con estadísticas clave como: tiempos promedio de entrega, zonas con retrasos frecuentes, desempeño de repartidores.	8	Porque requiere la interfaz gráfica interactiva y la lógica backend para consultar los datos necesarios.
RF_14	El sistema deberá integrar una pasarela de pagos que	8	Ya que requiere diseñar e implementar en la interfaz del cliente

Requisito		Estimación	Argumento
MUST	permita a los usuarios con el rol "Cliente" efectuar el pago correspondiente a sus pedidos de forma segura.		el flujo de pago con selección de método y manejo de estados de transacción, integrar y configurar la pasarela de pagos en el backend con validación de seguridad, manejo de respuestas y errores, registro de transacciones, además de pruebas end-to-end.
RF_15	El usuario de rol "Operador" deberá tener acceso a una interfaz para asignar los pedidos a un repartidor según su ubicación, esta interfaz cuenta con la ubicación de todos los repartidores.	8	Ya que requiere diseñar e implementar en la interfaz del operador un mapa interactivo que muestre en tiempo real las ubicaciones de los repartidores con filtros de proximidad y selección, así como desarrollar en el backend la lógica de consultas para determinar el repartidor más cercano
SHOULD			
RNF_1	El sistema debe contar con una interfaz responsive, web y móvil, que permitan su visualización en diferentes dispositivos y plataformas.	13	La complejidad de esta tarea es más determinada por el tech stack que se vaya a utilizar para realizar la aplicación pues herramientas como flutter flow permiten la portabilidad de la interfaz fácilmente.
MUST			
RNF_2	El sistema en su aplicación móvil deberá ser compatible tanto con dispositivos iOS como Android.	5	Ligado al RNF_1, esta tarea no resulta compleja si se utiliza una herramienta multiplataforma
MUST			
RNF_3	El sistema deberá contar con un ingreso autorizado de forma segura, verificando los datos del usuario y su rol.	3	Porque es una visual muy estándar al igual que la lógica backend usada para la misma.
MUST			
RNF_4	El sistema deberá ser resistente a ataques y otras vulnerabilidades de ciberseguridad comunes	13	Porque requiere hacer testeo extensivos y el proceso de pentesting puede ser demorado dada la cantidad de EndPoints que requiere la aplicación.
MUST			
RNF_5	El sistema deberá integrarse con la API de WhatsApp Business para	8	Porque requiere una integración con la API de Whatsapp y esto implica configurar credenciales y permisos, desarrollar la lógica para enviar y recibir mensajes según eventos del sistema, y realizar pruebas de funcionamiento y seguridad.
COULD			

Requisito		Estimación	Argumento
RNF_6 COULD	Las notificaciones de cancelación o modificación para los usuarios de tipo "Repartidor" no podrán exceder 5 segundos de delay en su envío.	5	Porque para cumplir esto la infraestructura y tecnologías usadas deben permitir una latencia menor a 5 segundos, pero esto es una latencia bastante alta.
RNF_7 SHOULD	El sistema en su aplicación móvil debe de poder funcionar en dispositivos de baja gama.	8	Ya que requiere optimizar el rendimiento de la app móvil en cuanto a consumo de memoria, peso, uso de red y renderizado, usando buenas prácticas de desarrollo. En gran medida está determinado por los frameworks utilizados por la app.
RNF_8 MUST	El sistema deberá poder mantener un mínimo 30 conexiones simultáneas con alta estabilidad y rendimiento	5	Ya que requiere configurar la infraestructura para manejar al menos 30 conexiones simultáneas con estabilidad, optimizar el backend para uso concurrente, gestionar recursos eficientemente y realizar pruebas de carga para asegurar el rendimiento sostenido. Pero igual está muy determinado por el framework del backend y proveedor de hosting del mismo.
RF_16 MUST	RF_16 (RQ04). El sistema deberá contar con una aplicación móvil para los roles "Repartidor" y "Cliente"	13	Ya que el requerimiento implica la creación de app móviles específicamente para esos roles, representa una historia grande, compleja, y con múltiples dependencias técnicas y de diseño. Considerando que ya están bien definidos los roles y sus respectivos servicios, se considera una tarea simple dependiendo de los frameworks que se usan.