



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá

Facultad de Ingeniería

Curso: Ingeniería de Software

**Estudiante: Sergio Tovar Vasquez, Emiliano Guerra,
Pablo Hernando Suarez Vargas, Julian David Murillo**

Rodriguez

Actividad en clase - Requerimientos

PUNTO 3) Requisitos/Requerimientos

- Sistematizar la recepción de pedidos desde Whatsapp o Email
- Cálculo estimado de entrega teniendo en cuenta información meteorológica y distancia
- Soporte para teléfonos
- Soporte para web
- Soporte para computador de escritorio
- Entregar la primera versión Funcional en el mes de junio o antes
- Realizar el desarrollo de la aplicación con un presupuesto entre 20 y 25 millones de pesos
- La aplicación no debe ser pesada y debe ser trabajable con una baja conectividad
- El sistema deberá organizar las rutas de cada entrega
- El sistema deberá poder ingresar pedidos de forma manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)
- El sistema deberá localizar al mensajero
- El sistema deberá comunicar en tiempo real la ubicación del mensajero a los operadores
- El sistema deberá ubicar los puntos relevantes a una entrega y su ruta por medio de un mapa
- El sistema deberá rastrear el estado de cada pedido y transmitirlo a los operadores
- El sistema deberá comunicar el estado del pedido constantemente a los operadores
- El sistema deberá comunicar el estado del pedido periódicamente a los clientes
- El sistema deberá asignar pedidos a mensajeros disponibles
- Definición de permisos roles (usuarios, clientes, Mensajeros, Administrativos)

- El sistema debe tener una vista (interfaz) para los clientes
- El sistema debe tener una vista (interfaz) para los Administrativos
- El sistema debe tener una vista (interfaz) para los Mensajeros
- Soporte para diferentes métodos de pago por medio de la aplicación
- Recepción de retroalimentación sobre la entrega por parte de clientes

Para un total de 23 requerimientos

PUNTO 4) División de requerimientos

Requisitos/Requerimientos funcionales	los puntos relevantes a una entrega y su ruta por medio de un mapa	(usuarios, clientes, Mensajeros, Administrativos)
-El sistema deberá organizar las rutas de cada entrega	-El sistema deberá rastrear el estado de cada pedido y transmitirlo a los operadores	-El sistema debe tener una vista (interfaz) para los clientes
-El sistema deberá poder ingresar pedidos de forma manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)	-El sistema deberá comunicar el estado del pedido constantemente a los operadores	-El sistema debe tener una vista (interfaz) para los Administrativos
-El sistema deberá localizar al mensajero	-El sistema deberá comunicar el estado del pedido periódicamente a los clientes	-El sistema debe tener una vista (interfaz) para los Mensajeros
-El sistema deberá comunicar en tiempo real la ubicación del mensajero a los operadores	-El sistema deberá asignar pedidos a mensajeros disponibles	-El sistema deberá recibir y procesar cambios en los pedidos por parte de clientes.
-El sistema deberá ubicar	-Definición de roles	-Soporte para diferentes métodos de pago por medio de la aplicación

Requisitos/Requerimientos no funcionales

- Sistematizar la recepción de pedidos desde Whatsapp o Email
- Cálculo estimado de entrega teniendo en

- cuenta información meteorológica y distancia
- Soporte para teléfonos
- Soporte para web
- Soporte para computador de escritorio

- La aplicación no debe ser pesada y debe ser trabajable con una baja conectividad
- Recepción de retroalimentación sobre la entrega por parte de clientes

PUNTO 5) Organización por MoSCoW

Must Have	Should Have	Could Have	Won't Have
El sistema deberá organizar las rutas de cada entrega	El sistema deberá poder ingresar pedidos de forma manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)	-La aplicación no debe ser pesada y debe ser trabajable con una baja conectividad	-Soporte para teléfonos
-El sistema deberá localizar al mensajero	El sistema deberá comunicar en tiempo real la ubicación del mensajero a los operadores	-Cálculo estimado de entrega teniendo en cuenta información meteorológica y distancia	-Soporte para computador de escritorio
-Sistematizar la recepción de pedidos desde Whatsapp o Email	El sistema deberá asignar pedidos a mensajeros disponibles	-Recepción de retroalimentación sobre la entrega por parte de clientes	El sistema deberá ubicar los puntos relevantes a una entrega y su ruta por medio de un mapa
El sistema deberá rastrear el estado de cada pedido y	El sistema deberá comunicar el estado del pedido		Soporte para diferentes métodos de pago por medio de la aplicación

transmitirlo a los operadores	constantemente a los operadores		
Soporte para web			
El sistema deberá poder ingresar pedidos de forma manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)			
Definición de roles (clientes, Mensajeros, Administrativos, debug)			
El sistema debe tener una vista (interfaz) para los clientes			
El sistema debe tener una vista (interfaz) para los Administrativos			
El sistema debe tener una vista (interfaz) para los Mensajeros			
El sistema deberá recibir y procesar cambios en los pedidos por parte de			

clientes.			
El sistema deberá comunicar el estado del pedido periódicamente a los clientes			

PUNTO 6): Prioridad Fibonacci

Funcionalidades	Puntos de historia	Argumentación
El sistema deberá organizar las rutas de cada entrega	8	Como no va a tener GUI durante éste sprint, la ruta puede ser resultado de una consulta a una API de maps, convertida a serie de instrucciones.
El sistema deberá localizar al mensajero	3	Al tener la información GPS del celular del mensajero, solo hay que considerar que ésta llegue a la base de datos
Sistematizar la recepción de pedidos desde Whatsapp o Email	13	La lectura, formateo y transporte de los datos de un pedido tiene muchos puntos de posible fallo y requiere conectar muchas cosas.
El sistema deberá rastrear el estado de cada pedido y transmitirlo a los operadores	5	El modelado de los estados para cada pedido y las notificaciones para cambios de estado hacen un trabajo sencillo pero largo
Soporte para web	40	Desplegar el sitio web y el trabajo de adecuación de frameworks y conexión con los dispositivos es trabajo multifacético y de alta complejidad.

El sistema deberá poder ingresar pedidos de forma manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)	3	Sólo implica la lectura de datos y trabajo de más alto nivel
Definición de permisos roles (clientes, Mensajeros, Administrativos, debug)	8	Es un largo trabajo de materializar los casos de uso, por lo tanto será de baja complejidad, pero largo.
El sistema debe tener una vista (interfaz) para los clientes	13	Se mostrará la ubicación de su pedido y un formulario para que puedan hacer nuevos pedidos
El sistema debe tener una vista (interfaz) para los Administrativos	20	Gracias a la cantidad de poderes que tienen, tendrá mucho trabajo de alta complejidad.
El sistema debe tener una vista (interfaz) para los Mensajeros	8	La información para mensajeros será solamente su ruta actual y una ventana de alertas o mensajes a enviar para administrativos
El sistema deberá recibir y procesar cambios en los pedidos por parte de clientes.	5	Debido a que los pedidos estarán modularizados el mayor trabajo de esta funcionalidad es un espacio para que los clientes puedan poner sus cambios
El sistema deberá comunicar el estado del pedido periódicamente a los clientes	3	Gracias a que depende de otras funcionalidades, sólo requiere conectar información entre los actores, si se decide utilizar una api de whatsapp, seguirá siendo sencillo.
El sistema deberá poder ingresar pedidos de forma	3	

manual desde la oficina (pudiendo importarse desde .xcl o archivos de texto)		Implica lectura de archivos estructurados y su transformación en registros del sistema. Es una funcionalidad simple, de alto nivel, y bien acotada, sin complejidad técnica alta.
El sistema deberá comunicar en tiempo real la ubicación del mensajero a los operadores	8	Aunque se puede usar un servicio de localización continua, la sincronización en tiempo real y el manejo de actualizaciones frecuentes puede ser demandante tanto en diseño como en implementación.
El sistema deberá asignar pedidos a mensajeros disponibles	13	Implica el diseño de un algoritmo de asignación basado en múltiples factores como disponibilidad, ubicación, tiempo estimado, etc. Es de alta complejidad.
El sistema deberá comunicar el estado del pedido constantemente a los operadores	5	Aprovechando el modelado de estados ya existente, se puede usar una suscripción o sistema de notificación interna para mantener informados a los operadores.
-La aplicación no debe ser pesada y debe ser trabajable con una baja conectividad	8	Requiere una optimización transversal del sistema: uso de recursos mínimos, almacenamiento local, sin perder funcionalidades clave.
-Cálculo estimado de	20	Involucra la integración con APIs de clima y mapas, además de lógica para

entrega teniendo en cuenta información meteorológica y distancia		estimar tiempos dinámicamente. Es compleja y propensa a errores.
-Recepción de retroalimentación sobre la entrega por parte de clientes	3	Consiste en el diseño de un formulario simple y su almacenamiento en la base de datos. Es sencillo y poco demandante.