

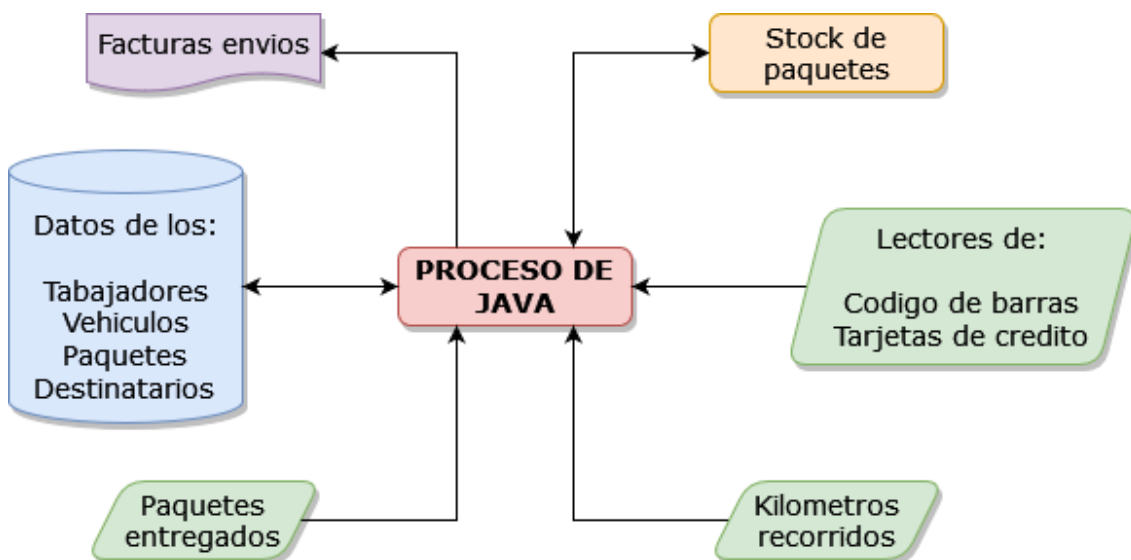
La empresa mocmoc desea una aplicación que posea una estructura rígida, para almacenar la información de sus envíos. Hemos optado por un modelo de ciclo de vida en cascada con retroalimentación.

La aplicación será de software libre, decidimos desarrollar el proyecto con el lenguaje de programación Java, el cual, al ser un lenguaje orientado a objetos podremos crear una aplicación lo bastante amplia para abarcar cualquier requerimiento solicitado por el cliente.

Así mismo, para realizar dicho proyecto, nos apoyaremos en una herramienta que nos permite trabajar con lenguaje Java, como NetBeans de Apache.

Tanto el lenguaje Java como el paquete NetBeans lo podemos descargar desde la web de Apache, y tras su instalación, ponernos a trabajar desarrollando la aplicación.

Para comenzar con el proyecto creamos un pequeño esquema indicativo de los distintos requerimientos solicitados:



## Fases del desarrollo de Software

1. Análisis de requisitos .....	2
2. Diseño.....	4
3. Codificación .....	5
4. Pruebas .....	5
5. Documentación.....	5
6. Explotación .....	6
7. Mantenimiento .....	6

# 1. Análisis de requisitos

Requisitos funcionales:

- Almacenar los datos de los trabajadores:
  - DNI
  - Nombre
  - Teléfono
  - Localidad
  - Dirección
  - Salario
  - N° Seguridad Social
- Almacenar los datos de los vehículos de transporte:
  - Matricula
  - Modelo
  - Tipo
  - Potencia
- Almacenar los datos de los paquetes:
  - Código
  - Descripción
  - Peso
  - Stock
- Almacenar los datos de los destinatarios:
  - DNI
  - Nombre
  - Teléfono
  - Dirección
  - Paquetes recibidos
- Proporcionar las facturas de los envíos.
- Llevar la cuenta de la cantidad de paquetes entregados por cada trabajador.
- Llevar la cuenta de los kilómetros recorridos por cada trabajador.
- Operar con lectores de código de barras y tarjetas de crédito.

Requisitos no funcionales:

- Reducir el tiempo de respuesta de la aplicación.

Tras estas premisas iniciales, planificaremos una serie de reuniones con el cliente para perfilar con más detenimiento cada uno de los requerimientos de la aplicación a diseñar, como, por ejemplo:

- Al necesitar emitir facturas, necesitamos igualmente incluir los datos de los clientes a los que irán dirigidas, y el código del vendedor que realizará la venta para poder controlar cuántas transacciones ha realizado dicho empleado.
- Para poder admitir tarjetas de crédito como pago, tendremos que poner un campo nuevo en la factura que permita indicar la forma de entrega económica y opte por simplemente indicar qué cantidad se abona en metálico o que se va a hacer uso del mencionado dispositivo magnético para realizar el pago.
- Los paquetes que se vayan a entregar se obtendrán de la tabla de stock de paquetes, pero se controlarán en una nueva tabla que recogerá en cada registro la factura a la que pertenece.

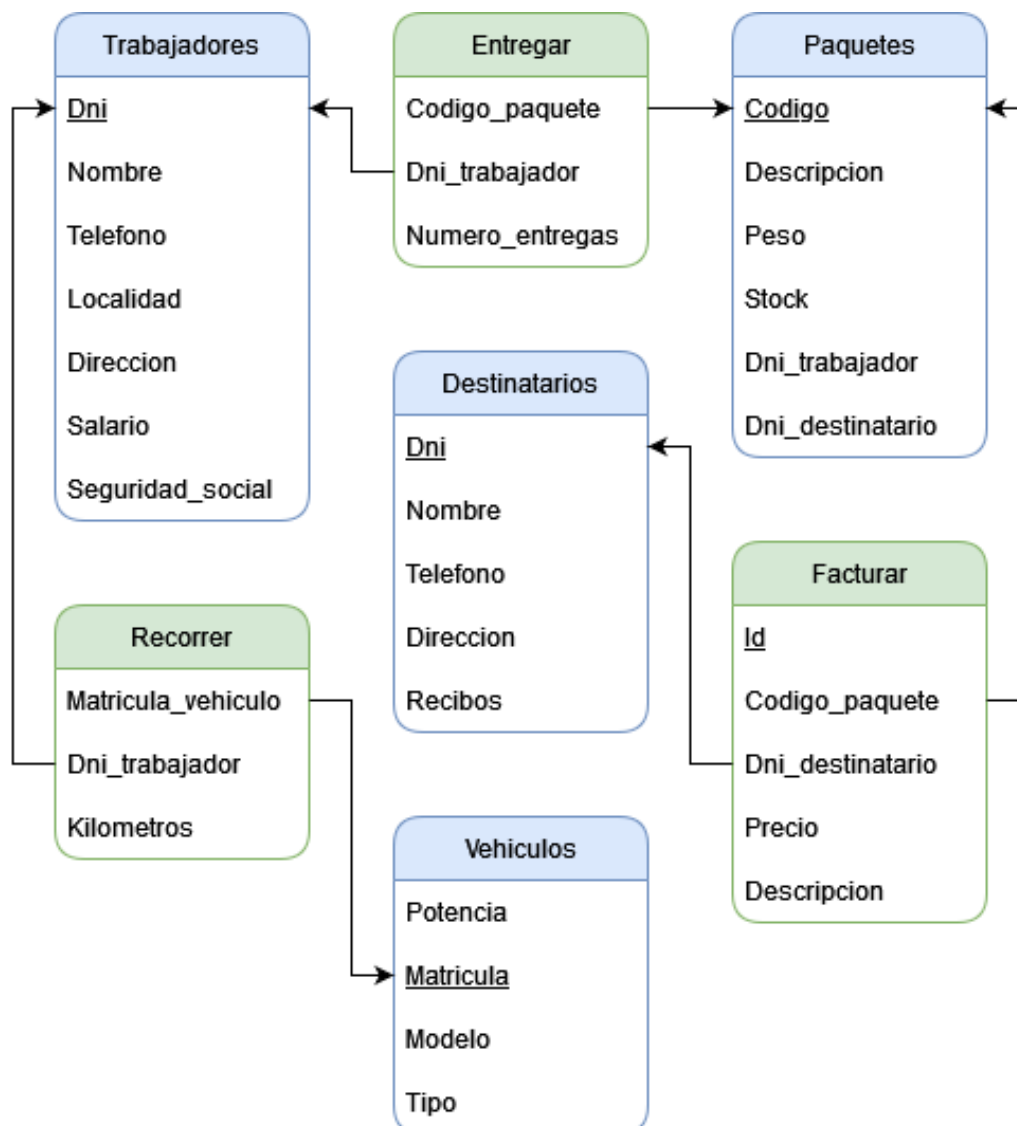
Cuando finalicemos con el refinamiento de todas estas premisas gracias a las reuniones con los encargados de la empresa y empleados que vayan a manejarla, sólo nos quedará comprobar cuáles son los equipos que se van a utilizar y si será necesario ampliar o modificar el parque informático de dicha empresa.

Con todo ello terminamos con la fase del análisis de requisitos y tendremos que enfocar el problema en base a un diseño eficaz, para lo cual podemos basarnos en el esquema realizado anteriormente.

## 2. Diseño

Crearemos una base de datos con SQL para poderla utilizar con Java mediante JDBC.

Tenemos cuatro entidades ya definidas con claridad como son trabajadores, vehículos, paquetes y destinatarios aunque, como vimos anteriormente, tendríamos que crear otras como son facturas con la que podríamos obtener los paquetes enviados, las facturas a emitir o ya emitidas, los trabajadores que han enviado el paquete, etc. factura en la que se grabará toda la información individual de cada paquete entregado, y destinatario para poder identificar a quién se le realiza la entrega e ir controlando los envíos realizados.



### 3. Codificación

Dado que el cliente ha decidido utilizar software libre, procederemos a la creación de la aplicación utilizando el lenguaje Java y apoyándonos en el entorno de desarrollo NetBeans, el cual usaremos para el diseño de la base de datos y toda la aplicación en general.

### 4. Pruebas

Obteniendo los ficheros objeto en la fase de codificación, podremos realizar las pruebas unitarias gracias a JUnit que es el entorno de pruebas para Java.

Una vez realizada con éxito este conjunto de pruebas unitarias, comprobaremos el funcionamiento de todo el sistema con todas sus partes interrelacionadas, para lo cual contactaremos con el cliente con el fin de poder implementar en sus equipos una versión beta que nos permita probar su desarrollo en el entorno donde se implantará definitivamente.

### 5. Documentación

Cuando todas las pruebas hayan finalizado con éxito, comenzaremos a crear la documentación donde especificaremos todas y cada una de las etapas realizadas en el desarrollo del software, es decir, tendremos que entregar tres guías al cliente:

- **Guía técnica:** Dirigida a analistas y programadores, nos permitirá realizar correcciones y mantener la aplicación en un futuro. Incluiremos el diseño de la aplicación, la codificación de los programas y las pruebas que se le han realizado.
- **Guía de uso:** Dirigida a los usuarios finales o clientes. Describiremos la funcionalidad de la aplicación, cómo comenzar a ejecutarla, ejemplos de uso, requerimientos software y solución a los posibles problemas que se puedan presentar, es decir, intentaremos dar a los usuarios finales toda la información necesaria para utilizar la aplicación.
- **Guía de instalación:** Incluiremos toda la información necesaria para la puesta en marcha y la explotación, así como la seguridad del sistema. Esta información la dirigiremos al personal informático responsable de la instalación, en colaboración con los usuarios finales. Con ella daremos toda la información para garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa.

## 6. Explotación

Comprobado que el software es fiable, carece de errores y terminado el proceso de documentación, procederemos a dar a conocer la aplicación a los usuarios finales para que comiencen a utilizarla, o lo que se denomina fase de explotación para lo cual haremos que los futuros clientes estén presentes en el proceso de instalación y le iremos comentando el proceso, realizando las beta test en los equipos del cliente y con carga de trabajo normales (como el software está diseñado para el cliente, nosotros realizaremos la instalación).

## 7. Mantenimiento

Para finalizar con el ciclo de vida del software que hemos creado procederemos a pactar con el cliente un servicio de mantenimiento de la aplicación que pueda conllevar una futura mejora de la funcionalidad del software, nuevas necesidades, expansiones o incluso modificaciones y actualizaciones para adaptarse a las nuevas tendencias del mercado, o al nuevo hardware que pueda adquirir el cliente.

Le ofreceremos al cliente un precio por un mantenimiento perfectivo (mejorará su funcionalidad), evolutivo (modificaciones o expansiones para futuras nuevas necesidades), adaptativo (modificaciones o actualizaciones que se adapten a las nuevas tendencias o componentes de hardware) y correctivos (para errores que pueda tener la aplicación en el futuro).