# TAREA CD 01

1. **Sintetiza el análisis de requerimientos del sistema para nuestro cliente. Plantea el diseño y determina el modelo de ciclo de vida más idóneo para esta aplicación.**

   **Análisis de requerimientos del sistema:**

   - Funcionales: Proporcionar facturas de las ventas, llevar la cuenta de lo que vende cada trabajador, controlar el stock de productos en almacén, operar con lector de código de barras y tarjetas de crédito, controlar los precios de los productos y ofrecer la posibilidad de operar con ellos, y almacenar información de sus trabajadores (DNI, nombre, apellidos, número de la Seguridad Social, fecha de nacimiento, teléfono y localidad) y de los productos (código, marca, nombre comercial, precio, cantidad).
   - No funcionales: El tiempo de respuesta de la aplicación ha de ser lo menor posible y no se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.

   **Diseño funcional-estructural:**

   - Productos: Almacenará la información de los productos y permitirá controlar los precios y operar con ellos.
   - Trabajadores: Almacenará información de los trabajadores y llevará la cuenta de las ventas de cada uno.
   - Ventas: Contendrá información detallada sobre las ventas y proporcionará facturas de las mismas. Además, cada vez que se produzca una venta se modificará el stock en almacén y se añadirá la venta a la cuenta del trabajador correspondiente.
   - Almacén: Contendrá el número de unidades de cada producto lo que permitirá controlar su stock. Se modificará dicho stock según las ventas que se produzcan.

El ciclo de vida más idóneo para esta aplicación es el **modelo en cascada con retroalimentación**, que además es uno de los modelos más utilizados. Este modelo es el idóneo para proyectos rígidos (pocos cambios y poco evolutivos) y con requisitos claros, como éste. Y además, una de sus ventajas es que se puede retornar a etapas anteriores para introducir modificaciones o depurar errores.

2. **Planifica la codificación, indicando el lenguaje de programación y las herramientas que usarías para la obtención del código fuente, objeto y ejecutable, explicando por qué eliges esas herramientas.**

   **Codificación:**

El **código fuente** se realizará empleando un **lenguaje de alto nivel** (lo cual permitirá que pueda ejecutarse en cualquier tipo de ordenador), **orientado a objetos** (cada parte de la estructura será un objeto, con atributos y métodos propios que podrán ser llamados por otros objetos), **imperativo** (describe paso a paso el conjunto de instrucciones que deben ejecutarse para variar el estado del programa y gestionar la tienda). La codificación de la aplicación se realizará por tanto en **Java** y tendrá **licencia abierta** (ya que la tienda desea trabajar con software libre). Este lenguaje de programación no es directamente ejecutable por la máquina, siendo necesario traducirlo.
Para que el tiempo de respuesta de la aplicación sea el menor posible y disfrutar de las ventajas de lenguajes compilados e interpretados, primero se realizará una **compilación** (mediante un compliador) a un **lenguaje intermedio** (bytecodes) obteniéndose el **código objeto**, y éste será

interpretado durante cada ejecución por la **máquina virtual** para obtener **código ejecutable** por la máquina.

3. **Planifica las restantes fases del ciclo de vida, indicando en cada una el objetivo que persigues y cómo lo harías.**

    - **Compilación:** Para que el código fuente codificado en un lenguaje de alto nivel sea ejecutable por la máquina. Se realiza mediante un **compilador o/y un intérprete**.
    - **Pruebas:** Para asegurar la validación y verificación del software construido (que es fiable y carece de errores). Se realizan **pruebas unitarias** (consisten en probar, una a una, las diferentes partes de software y comprobar su funcionamiento) y **de integración** (consisten en la puesta en común de todos los programas desarrollados una vez pasadas las pruebas unitarias de cada uno de ellos).
    - **Explotación:** Para que los usuarios finales conozcan la aplicación y comiencen a usarla. Se transfiere el programa al equipo del usuario cliente, **se instala, se configura**, se llevan a cabo los **Beta Test** (últimas pruebas que se realizan en los propios equipos del cliente y bajo cargas normales de trabajo) y se finaliza con la **producción normal** (La aplicación pasa a manos de los usuarios finales y se da comienzo a la explotación del software).
    - **Mantenimiento:** Para control, mejora y optimización del software. Se pacta un servicio de mantenimiento con el cliente para mejorar la funcionalidad, nuevos requisitos, resolver errores detectados, adaptarse a nuevas tendencias en el mercado… lanzando **nuevas versiones y actualizaciones** del mismo.
    - **Documentación:** Para dar toda la información a los usuarios del software, para poder acometer futuras revisiones y reutilización de los programas para otras aplicaciones. Es un proceso que se realiza durante todo el ciclo de la vida del software. Se realizan guías técnicas (dirigidas a analistas y programadores), guías de uso (dirigidas a clientes) y guías de instalación (dirigidas al personal informático responsable de la instalación).

4. **Indica el ciclo de vida que usarías.**

    **1)Análisis:** Requerimientos funcionales y no funcionales.
    **2)Diseño:** Estructura (división en partes o entidades), relaciones, funciones, elección de lenguaje de programación y de Sistema Gestor de Bases de datos. Generación de documento de Diseño de Software y de pruebas (sin detallar).
    **3)Codificación:** Obtención de código fuente en el lenguaje seleccionado que cumpla exhaustivamente los requisitos (fase de análisis) y el diseño de la aplicación (fase de diseño).
    **4)Compilación:** Obtención de código máquina a partir de código fuente (fase de codificación).
    **5)Pruebas:** Unitarias (generación de documento de procedimiento de pruebas) y de integración (generación de documento de procedimiento de pruebas de integración).
    **6)Explotación:** transferencia, instalación y configuración en el equipo del cliente.
    **7)Mantenimiento:** control, mejora y optimización de software (actualizaciones, versiones).
    **8)Documentación:** Constante durante todo el ciclo de vida. Generación de guías técnicas, de uso y de instalación.