1. Análisis de requerimientos del sistema y elección del modelo de ciclo de vida.

1.1. Análisis de requerimientos funcionales y no funcionales

En primer lugar, se ha procedido a elaborar el documento en el que se especifican los requisitos del software, tanto funcionales como no funcionales, y esto son el conjunto de funciones que deberá realizar la aplicación que vamos a desarrollar.

Funcionales	No funcionales
El sistema deberá proporcionar facturas de ventas	El tiempo de respuesta deberá ser el menor posible.
El sistema podrá llevar la cuenta de lo que vende cada trabajador	No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando a la vez.
El sistema permitirá controlar el stock de los productos del almacén	
El sistema permitirá operar con lector de código de barras y tarjetas de crédito	
Controlar los precios de los productos y tener la posibilidad de poder operar con ellos	
Almacenamiento de información de trabajadores: - Nombre - Apellidos - DNI - Número de la Seguridad Social - Fecha de nacimiento - Teléfono - Localidad	
Almacenamiento de la información de los productos: - Código - Marca - Nombre comercial - Precio - Cantidad	

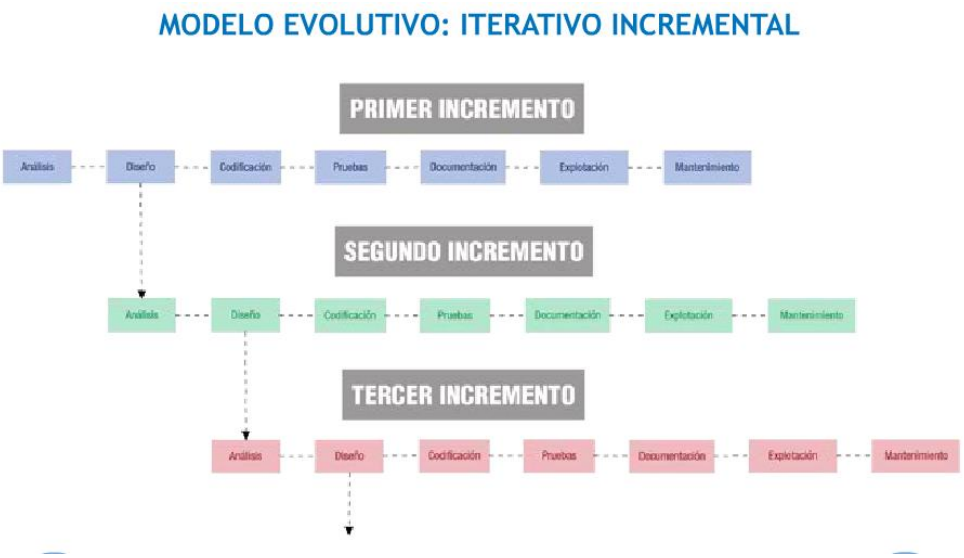
Una vez elaborado el documento de requerimientos funcionales y no funcionales del software, se procedería a hacer una serie de reuniones con el cliente para mostrarle cómo sería su aplicación y enseñarle el funcionamiento de esta.

1.2. Diseño y modelo de ciclo de vida.

En este proceso se definirá como se estructura el proyecto en sus diferentes partes estableciendo qué relaciones se establecerán entre estas.

Suponiendo que esta aplicación requerirá de actualizaciones posteriores cambiando a lo largo del tiempo, será necesario usar un ciclo de vida de modelo iterativo incremental, este modelo

permite una evolución del software a lo largo del tiempo. Este modelo está basado en el modelo en cascada con retroalimentación, donde cada fase sucede a la siguiente, pero con una retroalimentación entre etapas, de forma que podemos volver atrás en cualquier momento para corregir, modificar o depurar algún aspecto. Pero en este caso la aplicación se divide en subsistemas y en cada iteración se va generando versiones del software que se van refinando con cada nuevo incremento, de forma que cada versión obtenida será el punto de partida para la siguiente iteración que incluirá mayores funcionalidades hasta llegar a la versión del software más optima. A continuación, se ilustra en la siguiente imagen el modelo.



2. Fases restantes del ciclo de vida

2.1. Fase de codificación

El siguiente paso a realizar sería la codificación, y esto es el proceso de programación de nuestro software. Para esto, en primer lugar, elegiremos el lenguaje de programación que mejor se adaptaría. En este caso me decanto por utilizar el lenguaje de programación JAVA, ya que es el que mejor se adapta para la elaboración de este software.

En el proceso de codificación utilizaremos el framework de NetBeans, ya que este programa nos permite, por un lado, elaborar el código fuente de una forma limpia y más clara que con otros métodos tradicionales, obteniendo un código limpio y más fácil de trabajar; así mismo, este mismo programa se encargaría de la compilación del código fuente, que consiste en la traducción de este código fuente a un código objeto que se encontraría en lenguaje máquina (código binario), así mismo este programa (NetBeans) se encargaría de enlazar los archivos del código objeto para cargar un código ejecutable directamente en nuestra computadora.

2.2. Fase de pruebas

A lo largo de esta etapa se realizarán una serie de pruebas para verificar que se desarrollen correctamente todas las funciones de nuestro software que hemos creado anteriormente.

Esta fase se dividiría en dos partes:

- Pruebas unitarias: durante las cuales se verifican los módulos del software para comprobar su funcionamiento de manera independiente. En este caso usaremos JUnit como entorno de pruebas unitarias de Java (nuestro lenguaje de programación utilizado).
- Pruebas de integración: durante la cual se prueban todas las partes que están internamente relacionadas dentro de la aplicación una vez pasadas las pruebas unitarias de cada uno de ellos. Consiste en la puesta en común de todos los programas desarrollados.

Finalmente, se realizaría una prueba final a la que se denomina Beta Test, durante la cual el cliente interactúa con la propia aplicación ya terminada.

2.3. Fase de explotación

Una vez realizadas las pruebas y comprobado que todo el software funciona correctamente pasamos a la fase de explotación, durante esta fase se muestra a los usuarios finales la aplicación y estos comienzan a utilizarla.

Durante esta fase se procederá a la instalación y configuración del software en el/los equipos del cliente final. En esta fase conviene que los clientes estén presentes para poder ir informándolos acerca del funcionamiento de la aplicación. Una vez instalado el software, se procedería a configurarlo asignando los parámetros normales de la empresa. De esta forma, se aplicaría el Beta Test con los datos bajo una carga normal de trabajo en la empresa para comprobar que la aplicación se encuentra completamente operativa.

Una vez realizado esto, se pasaría a la fase de producción normal, donde la aplicación pasa a manos del cliente y se da comienzo a la explotación de esta.

2.4. Fase de Mantenimiento

Pasada la fase de explotación, lo siguiente que hay que hacer es el mantenimiento en el tiempo de este software. Durante esta fase, que se extiende en el tiempo, se realizan actualizaciones y cambios en el software, corrigiendo errores, implementando aquellas funcionalidades que se consideren necesarias e incorporando actualizaciones y mejoras que se adapten a las tendencias del mercado, leyes de protección de datos y a los posibles nuevos componentes del hardware de la empresa. Por ello se pactará con el cliente el precio del mantenimiento de este software y su duración en el tiempo.

2.5. Documentación

Esta fase se realizará a lo largo del todo el ciclo de vida del proyecto. Obteniendo documentos que informen de cada uno de los procesos que se han realizado. Lo que nos permitirá pasar de forma clara de una etapa a otra, y reciclar parte de algunos de estos programas para utilizarlos en otros proyectos.

A la luz de esto se elaborarán tres tipos de documentos:

- Guías técnicas (Dedicadas a personal técnico informático), donde queda reflejado:
    - El diseño de la aplicación
    - La codificación de los programas (lenguaje y código de programación utilizado)
    - Detalle de las pruebas que se han realizado para comprobar que todo funcione de manera óptima.

  El objetivo de esta guía es la de realizar las correcciones propias, permitir un mantenimiento futuro para las siguientes mejoras de la aplicación y facilitar el buen uso de este software.

- Guías de uso (Dedicadas a los usuarios finales/clientes de la aplicación), donde queda reflejado:
    - Describir la funcionalidad de la aplicación.
    - Los procesos para poder inicializarla.
    - Ejemplos de usos del programa.
    - Requerimientos del sistema para el uso de la aplicación.
    - Solución a posibles errores.

  El objetivo de esta guía es dar a los usuarios finales toda la información necesaria para que estos puedan utilizar la aplicación de forma correcta.

- Guía de instalación (Dedicadas al personal técnico informático encargado de la instalación), donde queda reflejado:
    - La puesta en marcha de la aplicación.
    - Como realizar la explotación del sistema.
    - Información sobre las medidas a tener en cuenta para garantizar la seguridad de la aplicación ante posibles amenazas.

  El objetivo de esta guía es garantizar la instalación de forma segura, confiable y precisa de nuestro programa.