

Análisis de requerimientos del Sistema:

El análisis de requisitos del sistema se divide en dos:

- **Requisitos funcionales:**
 - Proporcionar Facturas de venta
 - Llevar la cuenta de lo que vende cada trabajador
 - Control de stock de productos del almacén
 - Operar con lector de código de barras y tarjetas de crédito
 - Controlar los precios y ofrecer la posibilidad de operar con ellos
 - Almacenar información de los trabajadores:
 - DNI
 - Nombre
 - Apellidos
 - Número de la SS
 - Fecha Nacimiento
 - Teléfono
 - Localidad
 - Almacenar datos de los productos
 - Códigos
 - MARCA
 - Nombre comercial
 - Precio
 - Cantidad
- **Requisitos No Funcionales**
 - El tiempo de respuesta ha de ser lo menor posible
 - No se podrán procesar 2 peticiones a la vez aunque haya varios equipos funcionando

Es importante que el analista y cliente tengan buena comunicación para que la aplicación a desarrollar cumpla con las expectativas, así como para definir las reuniones periódicas, los mecanismos de actuación ante contingencias, etc

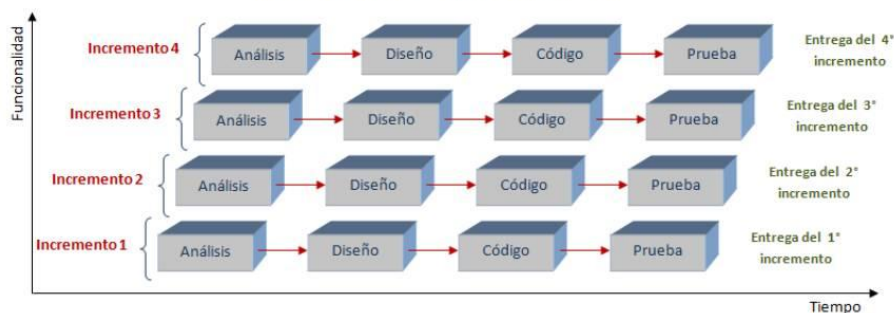
Diseño

En este punto ya se han definido los requisitos y el documento de análisis identifica como dividir el programa. Tras las reuniones con el cliente se puede proceder al diseño de la aplicación.

Como el cliente ha pedido usar herramientas de código abierto, usaremos Java como lenguaje de programación. Como gestor de bases de datos usaremos Oracle Database. Como IDE, NETBeans.

El modelo de ciclo de vida que decido escoger para este proyecto es el modelo evolutivo iterativo incremental, ya que tiene en cuenta la naturaleza cambiante y evolutiva del software. Es un modelo en cascada con realimentación, sus fases se repiten y se refinan y van propagando sus mejoras a las fases siguientes. En cada una de estas versiones se irán agregando funcionalidades, desde la más simple a la más compleja, hasta tener la totalidad del programa implementado.

Figura 4: Esquema del modelo Incremental.



Codificación

Durante esta fase se realiza el proceso de programación. Como ya he comentado se usará JAVA.

Se codificará toda la información y se llevará al código fuente. Esta tarea la hace el programador que deberá cumplir con todos los datos impuestos en las etapas de análisis y diseño.

Con el entorno de desarrollo NetBeans , al ser un entorno completo, cubriremos las tres partes de la codificación:

- Código Fuente: Será tarea de los programadores desarrollar este código en el entorno de desarrollo NetBeans utilizando el editor de código.
- Código Objeto: Utilizando el compilador de NetBeans obtendremos el código binario resultante de compilar el código fuente.
- Código ejecutable: Este código es el resultado de enlazar los archivos objeto, consta de un único archivo que puede ser ejecutado por el sistema operativo directamente. Este paso también lo realizaremos con la aplicación NetBeans.

Pruebas

En esta fase se realizan un conjunto de pruebas que aseguren la validación y verificación del software construido. Se distinguen dos partes:

- Pruebas unitarias:
 - Se prueba una a una las diferentes partes del software y se comprueba su funcionamiento. Como resultado de estas pruebas se genera un documento de procedimiento de pruebas. Con estos resultados se comparan con los resultados esperados de funcionamiento. Usaremos JUnit como entorno de pruebas.
- Pruebas de integración
 - Es la puesta en común de todos los programas desarrollados una vez pasadas las pruebas unitarias. Se genera un documento de procedimiento de pruebas de integración, que al igual que en las pruebas unitarias se compara los resultados obtenidos con los resultados esperables

Una vez que ya hemos realizado pruebas con el software para comprobar su fiabilidad y que carece de errores, continuamos con la fase de explotación.

Si todo esto es correcto, el software es fiable y carece de errores, se procede a su instalación en el equipo final del cliente, de ser posible con la presencia del mismo. Una vez instalado, comienzan los beta test, para realizar las últimas pruebas en los equipos finales y con cargas de trabajo normales.

Mantenimiento

En esta fase realizaremos los procesos de control, mejora y optimización del software, para ello pactaremos con el cliente los costes económicos y la duración del servicio de mantenimiento que mejor se adapten a sus necesidades.

El mantenimiento debe cubrir distintos cambios como pueden ser:

- Perfectivos: mejorar la funcionalidad del software.
- Evolutivos: cubrir las nuevas necesidades del cliente.
- Adaptativos: adaptarse a las nuevas tendencias del mercado o al nuevo hardware.
- Correctivos: corregir los errores que tenga en un futuro.

Documentación

En esta fase se desarrolla la documentación necesaria para que las etapas queden bien documentadas.

	Guía Técnica	Guía de Uso	Guía de Instalación
Aspectos	<ul style="list-style-type: none">-Diseño de la aplicación-Codificación de los programas-Pruebas realizadas	<ul style="list-style-type: none">-Descripción de la funcionalidad de la aplicación-Forma de comenzar a ejecutar la aplicación-Requerimientos software de la aplicación-Ejemplos de uso del programa-Solución a los posibles problemas que se puedan presentar	<ul style="list-style-type: none">-Puesta en marcha-Explotación-Seguridad del sistema
Objetivos	Facilitar un correcto desarrollo, realizar correcciones en los programas y permitir un mantenimiento futuro	Dar a los usuarios finales toda la información necesaria para utilizar la aplicación	Dar toda la información necesaria par garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa
Dirigido a	Personal técnico de informática (analistas y programadores)	Usuarios que van a usar la aplicación (clientes)	Personal informático responsable de la instalación