

Enunciado

La empresa BK ha recibido un nuevo encargo de software.

Se trata de diseñar una aplicación para una tienda especializada en vender productos estéticos.

La tienda desea trabajar con software libre. Además, desea explícitamente que la aplicación sea capaz de cumplir las siguientes tareas:

- ✓ 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.
- ✓ El tiempo de respuesta de la aplicación ha de ser lo menor posible.
- ✓ No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.
- ✓ La empresa también quiere almacenar información de sus trabajadores: DNI, nombre, apellidos, número de la Seguridad Social, fecha de nacimiento, teléfono y localidad. Asimismo, de los productos interesa almacenar: código, marca, nombre comercial, precio, cantidad.

Tendrás que diseñar una planificación del proyecto de desarrollo de ese software que cumpla con las premisas estudiadas en la presente unidad de trabajo.

Esencialmente, el proyecto se divide en los siguientes apartados:

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

1 ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA

Esta primera fase del proyecto es la de mayor importancia ya que todo lo demás dependerá de lo bien detallada que esté.

1.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE

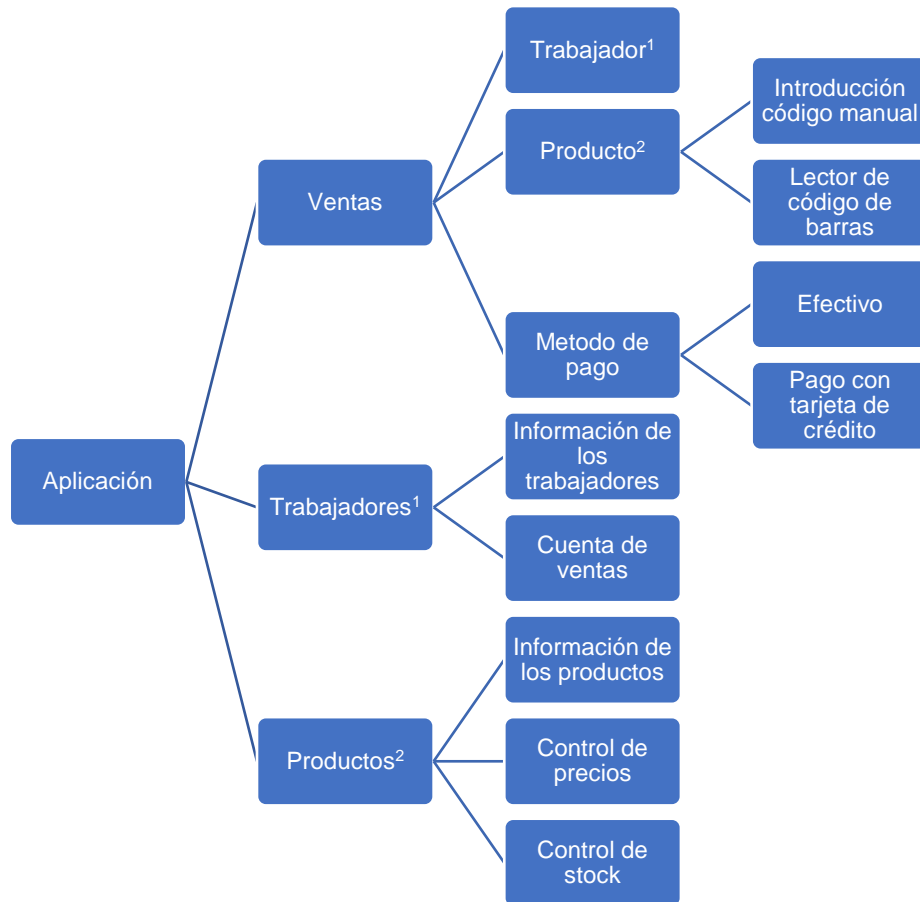
1.1.1 Requisitos funcionales

- a) Proporcionar facturas de ventas.
- b) Llevar la cuenta de lo que vende cada trabajador.
- c) Controlar el stock de productos en almacén.
- d) Operar con lector de código de barras y tarjeta de crédito.
- e) Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos.
- f) Almacenar información sobre los trabajadores.
 - 1) DNI
 - 2) Nombre
 - 3) Apellidos
 - 4) N° de la Seguridad Social
 - 5) Fecha de nacimiento
 - 6) Teléfono
 - 7) Localidad
- g) Almacenar información sobre los productos.
 - 1) Código
 - 2) Marca
 - 3) Nombre comercial
 - 4) Precio
 - 5) Cantidad

1.1.2 Requisitos no funcionales

- a) El tiempo de respuesta de la aplicación ha de ser el menor posible.
- b) No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.

1.2 DISEÑO DE ARQUITECTURA



1.3 COMUNICACIÓN CON EL CLIENTE

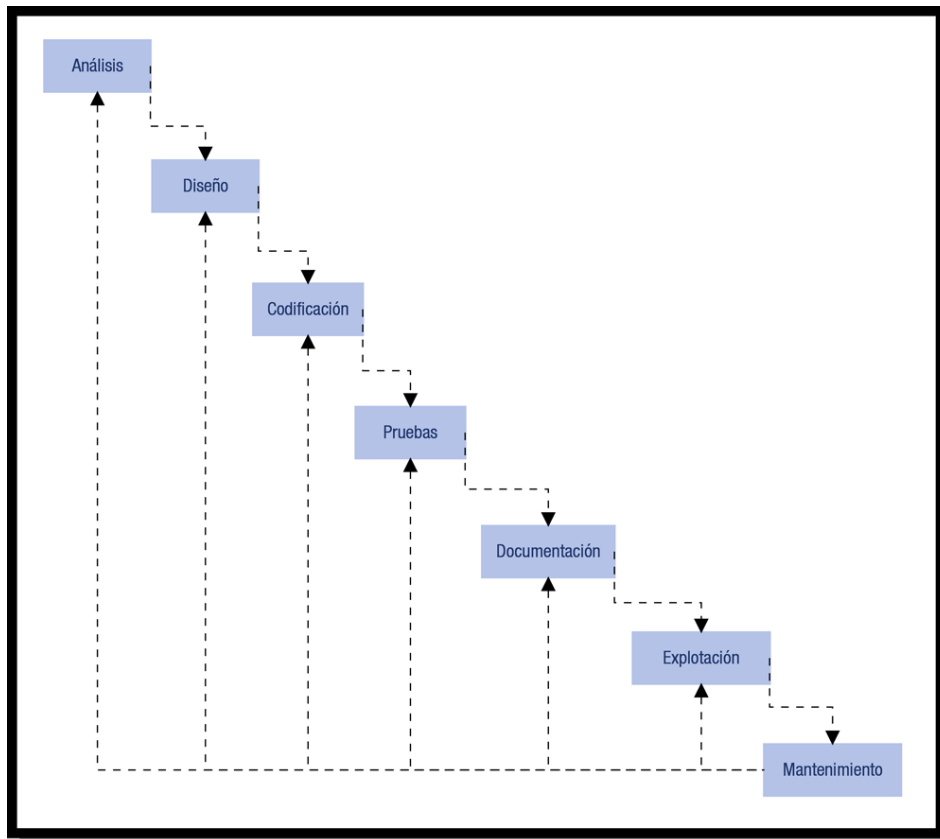
Tras las especificaciones iniciales para el diseño de la aplicación, se establecen con el cliente una serie de reuniones periódicas con el fin de precisar cada uno de los requerimientos a desarrollar y comprobar las características hardware. La comunicación entre cliente y analista es fundamental para desarrollar una aplicación que cumpla por completo con las expectativas del cliente.

2 DISEÑO

Una vez que tenemos claras las necesidades del cliente, podemos proceder con el diseño de la aplicación. Debemos crear un modelo funcional-estructural de los requerimientos del sistema global, para poder dividirlo y afrontar las partes por separado.

2.1 MODELO DE CICLO DE VIDA

El modelo de ciclo de vida mas adecuado para este proyecto considero que es el modelo en cascada con retroalimentación, ya que el proyecto es bastante rígido y se espera poca o nula evolución.



También se podría utilizar el modelo iterativo incremental, aunque no parece necesario sacar versiones del programa, ya que no se espera una evolución de este tras las indicaciones y necesidades del cliente.

2.2 DISEÑO DEL SOFTWARE

- ✓ El **lenguaje de programación** que vamos a utilizar es **Java**, ya que se adapta a las necesidades del cliente de ser un software libre y también a nuestras necesidades de programación.
- ✓ El **sistema gestor de Bases de Datos** que hemos decidido utilizar es **MySQL** ya que se puede utilizar como software libre y esto es uno de los deseos del cliente.
- ✓ Las **entidades y relaciones de la base de datos** serán las siguiente:



2.3 PLAN DE PRUEBAS

- ✓ Pruebas unitarias
- ✓ Pruebas de integración

3 CODIFICACIÓN

En este apartado codificaremos toda la información que hemos visto anteriormente y la llevaremos a código fuente. Esta tarea la realiza el programador, que además deberá cumplir con todos los datos impuestos en las etapas de análisis y diseño.

La codificación la vamos a realizar con el entorno de desarrollo NetBeans. Se trata de un entorno completo que nos permite cubrir las tres partes de la codificación.

- Código fuente:** Es tarea del programador/es desarrollar este código.
- Código objeto:** Utilizando el compilador de NetBeans, obtendremos el código binario resultante.
- Código ejecutable:** Este código se da como resultado de enlazar los distintos archivos objeto y consta de un único archivo que puede ser ejecutado por el sistema operativo.

4 PRUEBAS

Una vez obtenido el software, la siguiente fase es la realización de pruebas para asegurar la validación y verificación del software desarrollado.

4.1 PRUEBAS UNITARIAS

En esta parte comprobaremos el correcto funcionamiento de cada una de nuestras partes del software. Utilizaremos el entorno Junit para realizar estas pruebas.

4.2 PRUEBAS DE INTEGRACIÓN

En esta parte comprobaremos el correcto funcionamiento del software completo, con todas sus partes interrelacionadas.

4.3 BETA TEST

Tras la realización de las pruebas unitarias y de integración, se contactará con el cliente para llevar a cabo una última prueba en el entorno donde se va a utilizar el software por parte del cliente.

5 DOCUMENTACIÓN

En esta fase desarrollaremos toda la documentación necesaria para que todas las etapas queden perfectamente documentadas. Aunque, en realidad, la documentación se debe elaborar constantemente durante todo el ciclo de vida.

	Guía técnica	Guía de uso	Guía de instalación
¿A quién va dirigido?	Al personal técnico en informática	A los usuarios	Al personal informático responsable de la instalación.
Aspectos que quedan reflejados	<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 • Forma de comenzar a ejecutar la aplicación • Ejemplos de uso • Requerimientos software de la aplicación • Solución a los posibles problemas que puedan surgir 	Información necesaria para su puesta en marcha, explotación y seguridad del sistema
¿Cuál es su objetivo?	Facilitar un correcto desarrollo, realizar correcciones en los programas y permitir un mantenimiento futuro.	Que los usuarios finales tengan la información necesaria para la utilización de la aplicación	Dar toda la información necesaria para garantizar que la implantación de la aplicación se realice de forma

		segura, confiable y precisa
--	--	-----------------------------

6 EXPLOTACIÓN

Después de todas las fases anteriores, una vez que las pruebas nos demuestran que el software es fiable, carece de errores y hemos documentado todas las fases, el siguiente paso es la explotación, que es el proceso de instalación, puesta a punto y funcionamiento de la aplicación en el equipo final del cliente.

6.1 INSTALACIÓN

Los programas son transferidos al equipo del cliente y posteriormente configurados y verificados. Es recomendable que los clientes estén presentes y poder ir comentando todo lo que plantea la instalación.

6.2 CONFIGURACIÓN

Asignamos los parámetros de funcionamiento normal de la empresa y probamos que la aplicación es operativa.

6.3 FUNCIONAMIENTO

La aplicación pasa a manos de los usuarios finales y da comienzo la explotación del software.

7 MANTENIMIENTO

Se pactan con el cliente unos costes económicos y la duración del servicio para realizar procesos de control, mejora y optimización del software.

El mantenimiento debe cubrir distintos cambios como pueden ser:

- a) **Perfectivos:** Mejora de la funcionalidad.
- b) **Evolutivos:** Cubrir nuevas necesidades.
- c) **Adaptativos:** Adaptar el software a nuevas tendencias o hardware.
- d) **Correctivos:** Corregir errores.