

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:

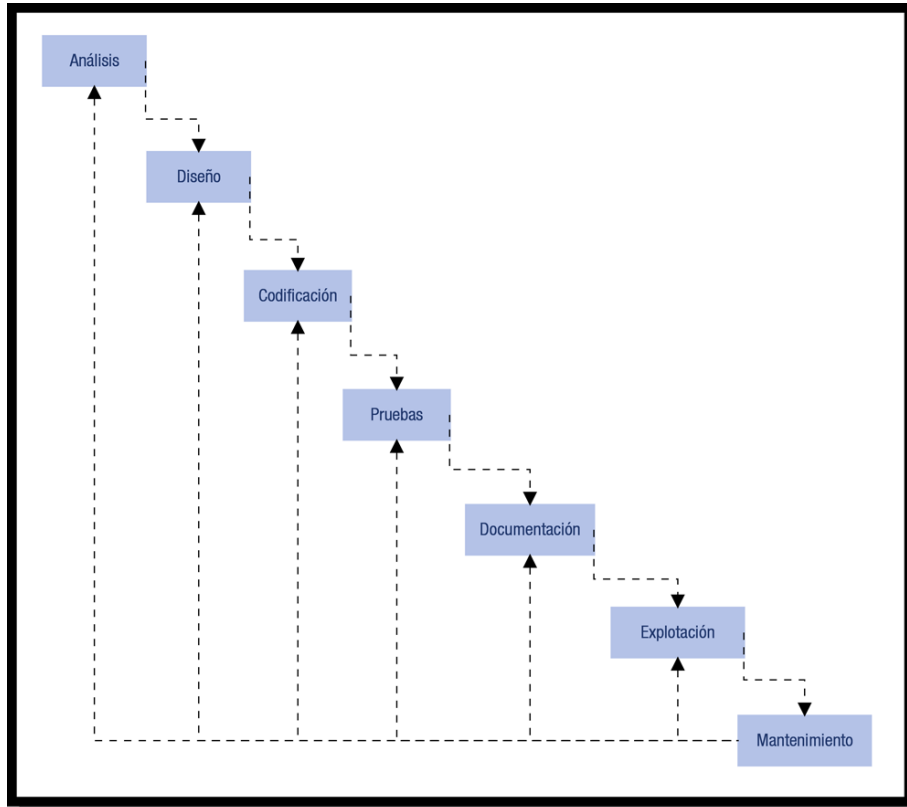
- Proporcionar facturas de las ventas.
- Llevar la cuenta de lo que vende cada trabajador.
- Controlar el stock de productos en almacén.
- Operar con lector de código de barras y tarjetas de crédito.
- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos.
- El tiempo de 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.

BK describe una aplicación especializada en vender productos estéticos que no requiere de muchos cambios y con una estructura rígida, por tanto, optaría por un modelo de ciclo de vida denominado **cascada con realimentación**. Es uno de los modelos más utilizados. Proviene del modelo anterior, pero se introduce una realimentación entre etapas, de forma que podamos volver atrás en cualquier momento para corregir, modificar o depurar algún aspecto.



Como dice que es software libre, utilizaría **Java** como lenguaje de programación de la aplicación, no solo por ser el lenguaje que estudiamos en esta formación, sino porque al ser orientado a objetos nos permite desarrollar el proyecto solicitado por el cliente. Esta función la desarrollaremos utilizando nuestro **entorno de desarrollo de Java** favorito (JDK, Eclipse, NetBeans), en este caso, utilizaríamos **NetBeans** ya que es libre y gratuito. Lo bueno de Java es que no causaría problemas de incompatibilidad, ya que Java no compila el archivo directamente, si no que compila nuestro código fuente con extensión .java en bytecodes (javac compiler), generando un código objeto con extensión .class que interpreta directamente la máquina virtual de java (JVM) instalada en nuestro Sistema Operativo, ya sea Mac, Windows o Linux.

En el documento especificación de requisitos software, constarán las tareas que BK desea explícitamente que la aplicación sea capaz de cumplir son las siguientes:

REQUISITOS **FUNCIONALES**:

- Almacenar los datos del trabajador:
 - DNI
 - Nombre
 - Apellidos

- Nº Seguridad Social
- Fecha de nacimiento
- Teléfono
- Localidad
- Almacenar información de los productos:
 - Código
 - Marca
 - Nombre comercial
 - Precio
 - Cantidad
- 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.

REQUISITOS **NO FUNCIONALES**

- 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.

Partiendo de estas premisas, nos reuniremos con el cliente para desarrollar un plan de acción sobre los puntos establecidos para el buen funcionamiento de nuestra aplicación, de forma que comentaremos posibles situaciones que se puedan presentar y que sea necesario abordar, como por ejemplo:

- Control de stock del almacén, conseguir que el programa nos avise cuando no haya stock mínimo o cuando se vaya a rebasar. De la misma manera, cuando no haya stock de un producto, se podría implementar un aviso al cliente por correo electrónico que avise cuando vuelve a estar disponible.
- Al emitir facturas necesitamos 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. De esta manera se controla la producción del vendedor y también habría que estudiar la Ley de Protección de Datos a la hora de almacenar datos de clientes.
- Habría que añadir un campo nuevo en la factura que permita indicar el método de pago.
- Los artículos que se vayan a vender se obtendrán de la tabla de artículos que indique la disponibilidad, el precio, el código y el descuento del mismo pero se controlarán en una nueva tabla que recogerá en cada registro la factura a la que pertenece.

Cuando se hayan perfilado los puntos a afrontar habría que averiguar los distintos equipos encargados de manotear el preciso funcionamiento de la tenacidad y renovar o modificarlos si fuese necesario, dando por concluido el **Análisis de Requisitos**.

Crearemos una base de datos con SQL para poderla utilizar con Java mediante JDBC, mediante la cual definiríamos las entidades Trabajadores y Artículos. Más adelante añadiríamos otras como Ventas para monitorear los artículos vendidos o las facturas. Con ello terminamos la fase del **Diseño** y comenzamos con la **Codificación**, y 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.

Cuando hayamos terminado de programar con éxito, crearemos la **documentación** donde especificaremos todas y cada una de las etapas realizadas en el desarrollo del software:

- **Guía técnica:** Dirigida a analistas y programadores, y que 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.

- **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 (clientes). Con ella intentaremos dar toda la información para garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa.

Cuando se verifique que el software sea confiable, libre de errores y se complete el proceso de documentación, continuaremos informando a los usuarios finales sobre la aplicación para que puedan comenzar a usarla, a esto se le llama **fase de explotación**, y para ello tendremos futuros clientes probando la versión beta del producto y el descubrimiento de posibles errores o mejoras que se subsanarían el consecuente **fase de mantenimiento**, donde nuestro producto podría contar con nuevas modificaciones y características adaptándose a las exigencias tecnológicas del momento.