

## Tarea 1

### 1.- 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:

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

### 1. Análisis

La empresa BK desea una aplicación para su tienda especializada en vender productos estéticos, por lo que las especificaciones que nos da el cliente es una estructura bastante rígida y lo suficientemente clara, por lo que el modelo de ciclo de vida que vamos a utilizar es **cascada con retroalimentación**.

Uno de los requisitos que nos pide es trabajar con software libre, así que una vez hablado con el cliente, se ha decidido que el proyecto se desarrollará en Java, concretamente con la OpenJDK 11, que es un lenguaje orientado a objetos y multiplataforma. Por lo tanto, inicialmente no es ningún problema que funcione en diferentes ordenadores con distintos sistemas operativos. Además Java es un lenguaje de programación muy escalable, de modo que si en un momento dado el cliente quiere escalar el software para que se pueda usar en dispositivos móviles, en tablet o mediante entorno Web, con Java no tendrá problemas.

Para desarrollar el proyecto utilizaremos el Entorno de Desarrollo Apache Netbeans, que también es software libre y trae implementados plugin para desarrollar los entornos gráficos de manera muy visual y cómoda, dando la posibilidad, llegado a la fase del desarrollo de diseñar junto con el cliente el aspecto que tendrá la aplicación, consiguiendo así más integración del cliente en el proyecto.

Como sistema de almacenamiento de los datos, usaremos la base de datos MySQL que se puede implementar tanto en un equipo local como en un servidor de una red privada y además también es software libre.

#### ➤ **Requisitos funcionales**

- Información de los trabajadores:
  - DNI
  - Nombre
  - Apellidos
  - Número de la Seguridad Social
  - Fecha de nacimiento
  - Teléfono
  - Localidad
- Productos:
  - Código
  - Marca
  - Nombre comercial
  - Precio
  - Cantidad.
- Proporcionar facturas de las ventas.
- Operar con lector de código de barras y tarjetas de crédito.
- Controlar el stock de productos en almacén.
- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos.
- Llevar la cuenta de lo que vende cada trabajador.

#### ➤ **Requisitos No funcionales**

- El tiempo de respuesta de la aplicación ha de ser lo menor posible.
- No se podrán procesar dos peticiones al mismo tiempo, aunque haya varios equipos funcionando simultáneamente.

- Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

Una vez que tenemos claro los pasos iniciales, lo siguiente es mantener una serie de reuniones con el cliente para conocer más en profundidad el funcionamiento de la empresa (Lógicamente lo concerniente a la aplicación que tenemos que desarrollar) y de esas reuniones obtenemos que la aplicación necesita algunas cosas más que los requerimientos iniciales del cliente.

- Para el control de stock del almacén se tiene que añadir un campo más a los Productos y el un Control de Stock Mínimo, de tal manera que cuando la aplicación de manera automática detecte que Cantidad es igual o inferior que el Stock mínimo, salte un aviso para que se reponga esos productos o en breve se romperá el Stock.
- Para las facturas de las ventas necesitamos también almacenar los datos de los clientes para las que van dirigidas, por lo tanto habrá una sección de los datos de los clientes.
- En las facturas, para cumplir uno de los requisitos de saber lo que vende cada vendedor, incluiremos el código del vendedor (Trabajadores) que hizo la operación.
- También se añadirá un campo con la forma de pago, el cual debe indicar si la compra ha sido realizada con tarjeta de crédito y por lo tanto se va a usar el sistema de pago con tarjetas, o la compra es en metálico.
- La factura se dividirá en dos partes, por un lado la factura en sí con los datos que se han mencionado anteriormente, y por otro irá el detalle de la factura, que serán los productos que se han vendido.
- En Artículos o Productos para llevar un control sobre el precio y poder operar sobre ellos, el precio lo modificaremos y crearemos por precio de coste y por un precio venta público.
- En el detalle de la factura, es decir en los artículos que se venden, añadiremos un campo de Descuento, para que en función de los intereses de la empresa en determinadas ventas, se pueda aplicar un descuento al cliente sobre un producto concreto.

Una vez analizada la empresa y vistas las necesidades del software a desarrollar, analizamos el sistema informático que tienen, para ver si puede utilizarse añadiendo alguna modificación, o directamente hay que cambiar los equipos para que, en base a todos los requisitos funcione perfectamente el software.

## 2. Diseño

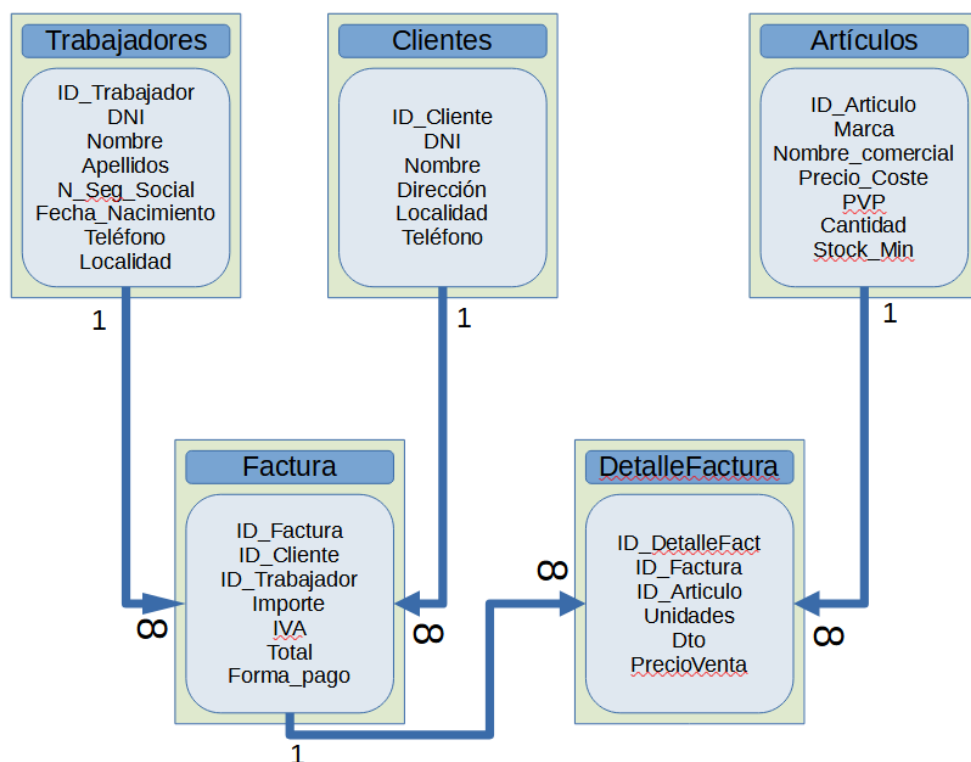
En base a lo analizado y con el objeto de crear un diseño eficaz, dividiremos el sistema en partes, y a cada una de ellas le estableceremos la relación que tienen unas con otras.

Como he dicho anteriormente, y debido a un requerimiento del cliente, el lenguaje de programación usado para el desarrollo del proyecto será Java, concretamente su versión Open Source OpenJDK 11.

El sistema gestor de base de datos a utilizar será MySQL, es la base de datos de código abierto número uno del mundo.

Para la definición de las entidades, por un lado tenemos a los **Trabajadores** y los **Artículos**, pero como hemos visto necesitamos crear otras entidades en la base de datos, tendríamos que crear **Cientes** como se para poder identificar a quien se realiza la venta, y poder tener un control de las compras que realizan los clientes. Y la otra entidad es la de Facturas que estará dividida en dos entidades directamente relacionadas, como es la **Factura** que contiene los datos del vendedor, del cliente, sistema de cobro, etc y la otra entidad llamada **DetalleFactura** en la cual tendremos los artículos adquiridos en esa factura.

En el siguiente esquema se muestran todos los campos que tendrá cada entidad y las relaciones entre las diferentes tablas:



Diseñaremos un Plan de pruebas que a posteriori será aplicado en la fase correspondiente así como un estilo de programación, para este tipo de proyecto se usara el patrón Modelo-Vista-Controlador que se explicará en la siguiente sección.

### **3. Codificación y Compilación**

Como he dicho anteriormente, y debido a un requerimiento del cliente, se utilizará Java, debido a su calidad de software libre, por eso se optó por la versión OpenJDK 11, que además es uno de los lenguajes más utilizados en el Mundo.

Como también se ha mencionado, el estilo para desarrollar la aplicación será usando en todo el proyecto el patrón de programación Modelo-Vista-Controlador, ya que con este modelo seguimos avanzando en la división de los problemas que iniciamos en la fase anterior, ya que cada sección de la aplicación estará dividida en otras partes. Por un lado, el Modelo que será el encargado de gestionar el acceso a la base de datos y por otro la Vista, que se encargara únicamente de gestionar la parte visual, es decir, de relacionarse directamente con el usuario. Por último, será el Controlador el encargado de repartir el trabajo entre el modelo y la vista.

Una exigencia del cliente es que no se puedan procesar dos peticiones al mismo tiempo, cosa que al tratarse de una base de datos el sistema podría hacer sin problemas. Debido a lo cual, se implementará en el Modelo (Acceso a los datos) para que solo se pueda acceder de una sola vez.

Para programar en Java usaremos el entorno de desarrollo Apache Netbeans, el cual tiene integradas todas las herramientas necesarias para la conexión con la base de datos. Tiene un entorno muy amigable para el desarrollo de las diferentes pantallas con las que va a tratar el usuario final. Como se mencionó en el Análisis, y gracias al ciclo de vida que estamos utilizando, volveremos a contactar con el cliente para que vea el resultado y si es necesario modificarlo o adaptarlo el estilo visual que tendrá la aplicación y adaptarse mejor a sus necesidades.

Durante la fase de Codificación se irá documentando cada parte y cada método, en un documento técnico de codificación, de modo que si en el futuro es necesario modificarlo podrá hacerse de forma rápida y ordenada.

### **4. Pruebas**

En el Diseño se estableció un Plan de pruebas, el objetivo de dicho plan es la validación y verificación del software creado. Para ello lo primero que haremos es que cada parte del proyecto sea sometido a pruebas unitarias, dichas pruebas serán realizadas mediante la librerías JUNIT. Y una vez terminadas las pruebas con éxito, se generara un documento con las pruebas realizadas y los resultados de las mismas.

Pasadas las pruebas unitarias procedemos a las de integración, mediante las cuales comprobaremos el funcionamiento de todo el sistema, además de establecer si

las diferentes partes están correctamente interrelacionadas. Una vez superado todas las pruebas, creamos el documento describiendo las pruebas de integración.

Llegados a este punto nos pondremos nuevamente en contacto con el cliente, para comenzar con él la verificación del producto. Para ello crearemos una compilación de la aplicación en su primera versión BETA, con lo cual el cliente podrá probar de manera controlada una versión totalmente operativa de la aplicación creada.

## **5. Documentación**

Antes de pasar a la fase de explotación, y dado que a lo largo de todas las fases del ciclo de vida hemos ido desarrollando diferentes documentos, se debe realizar una integración de todos ellos mediante tres guías:

- **Guía técnica**: es un documento dirigido únicamente para analistas y desarrolladores, el cual les permitirá realizar correcciones y mantener la aplicación en un futuro. En esta guía incluiremos todo el diseño de la aplicación, la codificación del proyecto, las pruebas y los resultados de las mismas.
- **Guía de uso**: destinada a los usuarios finales, es decir, a los trabajadores del cliente. Se describe la funcionalidad de la aplicación, como usarla con diferentes ejemplos, los requerimientos de hardware que necesita y soluciones a errores que se puedan presentar.
- **Guía de Instalación**: Este documento está dirigido a personal informático, es decir, a los responsables de hacer las instalaciones. En ella se incluye toda la información necesaria para la puesta en marcha de la aplicación, así como información sobre la seguridad del sistema.

## **6. Explotación**

Una vez verificada la aplicación cumpliendo con las exigencias del cliente, comprobado mediante las diferentes versiones Alpha que el software carece de errores, y terminada toda la documentación, es el momento de llevar a cabo las Beta Test mediante la implementación de la aplicación en los equipos del cliente. Y que sea utilizada con total normalidad, pero siendo aún supervisado por los desarrolladores, para comprobar que todo funciona correctamente.

Una vez pasado con éxito la fase Beta Test es el momento de pasar a producción final del proyecto, y se realiza el compilado final donde se dará al cliente tanto el software final como la documentación correspondiente.

## **7. Mantenimiento**

El último paso del proceso es el Mantenimiento, para lo cual pactaremos con el cliente un servicio adecuado a sus necesidades. Por un lado se realizará el mantenimiento correctivo sin coste alguno, para corregir los posibles fallos que puedan

surgir, y por otro lado se dará servicio ante una posible ampliación y mejoras en la funcionalidad, o la adaptación a tecnologías móviles, etc.

Se le ofrecerá un listado de precios con las posibles opciones de servicio, se opta por diferentes precios ya que no es lo mismo ampliar una funcionalidad que hacer una implementación del programa para dispositivos móviles como tablet.

Las diferentes opciones se son las siguientes:

- Mantenimiento Perfectivos, ante posible mejoras funcionalidades del software, como por ejemplo, que se pueda hacer entradas de datos en dos o más equipos al mismo tiempo. Aunque esto podría parecer que es una mejora, lo cierto es que requiere hacer un cambio mínimo en el Modelo de acceso a los datos.
- Mantenimiento Evolutivos, para posibles mejoras en el software, como por ejemplo, que su aplicación también pueda registrar las devoluciones de productos.
- Mantenimiento Adaptativos, este será el que más coste pueda tener y el precio no será cerrado, ya que dependerá de las necesidades del cliente, como adaptarse a nuevas tendencias de mercado (Dispositivos móviles), servicio de compras online, etc.