



## Entorno desarrollo de software

Aplicaciones Web (Universidad Complutense de Madrid)



Escanea para abrir en Studocu

# Desarrollo de software

## 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.

El modelo de ciclo de vida elegido será en **cascada con retroalimentación** puesto que los requisitos de la aplicación están claros y el proyecto es rígido (pocos cambios, poco evolutivo). Con este método, si es necesario se puede volver atrás en cualquier momento y corregir, modificar o depurar aspectos necesarios.

Otra opción, un poco más compleja, sería aplicar como ciclo de vida, el **modelo iterativo incremental**. Éste está basado en el modelo en cascada con realimentación, donde las fases se repiten y refinan, y van propagando su mejora a las fases siguientes.

Una de las especificaciones del cliente es que desea utilizar software libre. Teniendo en cuenta que cualquier código fuente, debe de presentar características como modularidad, corrección, fácil de leer, eficiencia y portabilidad, una buena elección es usar como lenguaje de programación **Java**, que además de facilitar la consecución de las características anteriormente expuestas se trata de un software libre y permite programación orientada a objetos, que nos

dará la suficiente libertad para poder crear una aplicación lo bastante amplia para abarcar cualquier requerimiento solicitado por el cliente.

Al tratarse un lenguaje ampliamente difundido, también dispone de una amplia gama de librerías que puede facilitar mucho nuestro trabajo.

Como entorno de desarrollo se puede usar **NetBeans** ya que es una herramienta consolidada en el mercado.

### **Análisis de Requisitos del Sistema:**

Como **requisitos funcionales** (funciones que tendrá que realizar la aplicación) tenemos los siguientes:

- 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.
- Almacenar los datos de los empleados (trabajadores):
  - DNI
  - Nombre
  - Apellidos
  - Nº Seguridad Social
  - Fecha de nacimiento
  - Teléfono
  - Localidad
- Almacenar información de los productos:
  - Código
  - Marca
  - Nombre comercial
  - Precio
  - Cantidad

Como **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.

Se planificará una serie de reuniones con el cliente para evitar requisitos mal planteados o que se produzcan contradicciones.

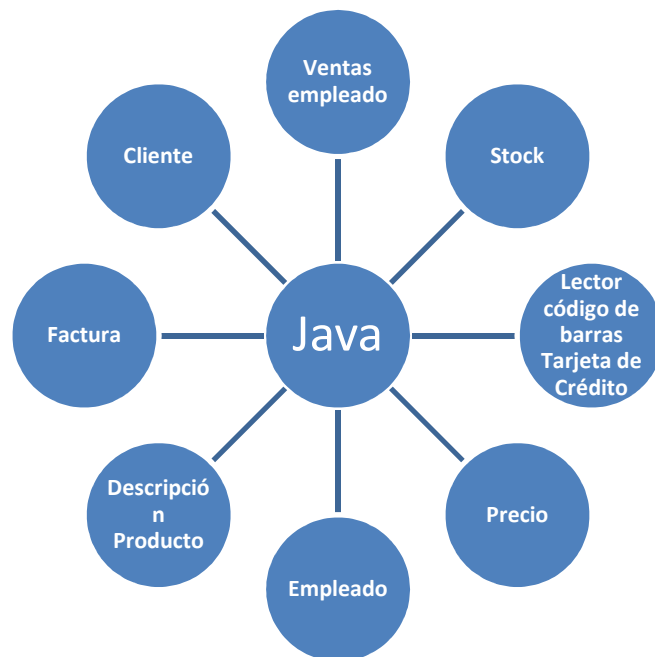
Se le puede sugerir una serie de mejoras, que incluyan por ejemplo, incluir los datos de los clientes para facilitar el sistema de facturación.

Puesto que el cliente especifica que se quiere controlar el stock, se añadirá un campo para controlar el stock mínimo de un producto y que de una alerta para un mejor control por parte del usuario final.

## Diseño:

En esta etapa vamos a dividir el sistema en partes y a determinar la función de cada una de ellas. Debemos determinar que hará exactamente cada parte, para ello debemos crear un modelo funcional-estructural de los requerimientos del sistema global y poder así dividirlo y afrontar las distintas partes por separado.

Nos creamos un pequeño esquema indicativo de los distintos requerimientos solicitados ampliado según nuestras sugerencias:



Basándonos en el esquema, se divide el sistema en partes, determinando la función que llevará a cabo cada una de ellas.

Crearemos una base de datos con SQL para poderla utilizar con Java.

Tenemos dos entidades ya definidas con claridad como son **Empleados** y **Productos**.

Tendríamos que crear otras como son **Ventas** con la que podríamos obtener los productos vendidos, las facturas a emitir, los empleados que han realizado la venta, etc.

**Factura** en la que se grabará toda la información individual de cada producto vendido, y **Cliente** para poder identificar a quién se le realiza la venta e ir controlando los pedidos realizados.

### Gestión de Empleados.

Debe de realizar todas las acciones relacionadas con la gestión de los productos (alta, baja, modificación, consulta). La información de los productos se guardará en una tabla de la base de datos con los siguientes campos:

idEmpleado, nif, nombre, apellidos, numSeguridadSocial, fechaNacimiento, telefono, localidad, idVenta.

### Gestión de Productos.

Debe de realizar todas las acciones relacionadas con la gestión de los productos (alta, baja, modificación, consulta). La información de los productos se guardará en una tabla de la base de datos con los siguientes campos:

idProducto, nombre, descripcion, marca, precioCompra, precioVenta, stockMinimo, cantidad.

### Gestión de Clientes.

Debe de realizar todas las acciones relacionadas con la gestión de los clientes (alta, baja, modificación, consulta). La información de los clientes se guardará en una tabla de la base de datos con los siguientes campos:

idCliente, nif, nombre, apellidos, direccion, telefono.

### Gestión de Ventas.

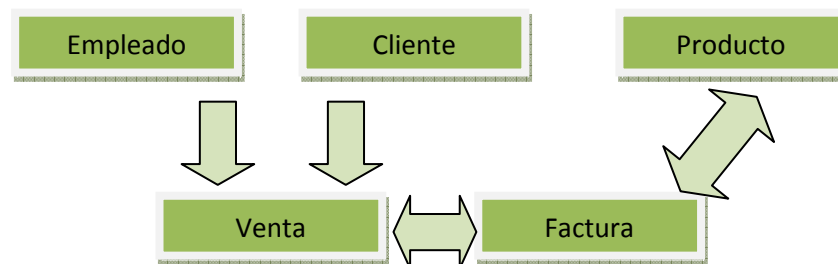
Debe de realizar todas las acciones relacionadas con la gestión de los pedidos (alta, baja, modificación, consulta). La información de los pedidos se guardará en una tabla de la base de datos con los siguientes campos:

idVenta, idCliente, fecha, idEmpleado, importeTotal, cobrada.

### Gestión de Facturas.

Debe de realizar todas las acciones relacionadas con la gestión de los pedidos (alta, baja, modificación, consulta). La información de los pedidos se guardará en una tabla de la base de datos con los siguientes campos:

idFactura, idVenta, cantidad, idProducto, precio, descuento, formaPago.



### Codificación:

Para facilitar la futura expansión de la empresa se optará por usar un **lenguaje de programación multiplataforma** como ya se mencionó al principio del documento. Como **sistema gestor de base de datos**, se puede usar **Microsoft® SQL Server® 2014 Express**, puesto que se puede descargar de forma gratuita. Es un software potente, robusto y estable, que cubre las necesidades del cliente, sin aumentar los costos.

Se procede a la creación de la aplicación utilizando el lenguaje Java y con el entorno de desarrollo JDeveloper, que se usará para el diseño de la base de datos, sus tablas, relaciones, y toda la aplicación en general, obteniendo los ficheros objeto en esta fase de codificación.

## Pruebas:

La realización de pruebas es imprescindible para asegurar la validación y verificación del software construido. En esta etapa se prueban los programas para detectar errores y se depuran.

El fichero objeto se usa para realizar las **pruebas unitarias** mediante JUnit que es el entorno de pruebas para Java.

Una vez realizada con éxito las pruebas unitarias, realizaremos las **pruebas de integración**. Comprobaremos el funcionamiento de todo el sistema con todas sus partes interrelacionadas, para lo cual contactaremos con el cliente con el fin de poder implementar en sus equipos una versión *Beta* que nos permita probar su desarrollo en el entorno donde se implantará definitivamente.

El período de prueba será pactado con el cliente.

## Documentación:

Todas las etapas en el desarrollo de software deben quedar perfectamente documentadas. Es necesario dar toda la información a los usuarios del software que hemos desarrollado y poder acometer futuras revisiones del proyecto.

Se debe de ir documentando el proyecto en todas las fases del mismo, para pasar de una a otra de una forma clara y definida. De esta forma crearemos los siguientes documentos:

- **Guía técnica.**

El objetivo es facilitar un correcto desarrollo, realizar correcciones en los programas y permitir un mantenimiento futuro. Se especificará el diseño de la aplicación, la codificación de los programas y las pruebas realizadas. Irá dirigido al personal técnico en informática (analistas y programadores).

- **Guía de uso.**

Con este documento buscamos dar a los usuarios finales toda la información necesaria para utilizar la aplicación. Contendrá la descripción de la funcionalidad de la aplicación, la forma de comenzar a ejecutarla, ejemplos de uso del programa, los requerimientos software de la aplicación y la solución de los posibles problemas que se pueden presentar.

Este documento irá dirigido a los usuarios que van a usar la aplicación (clientes).

- **Guía de instalación.**

A través de este documento se pretende dar toda la información necesaria para garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa. Contendrá toda la información necesaria para poner en marcha la aplicación, para su explotación y para mantener la seguridad del sistema.

Este documento estará dirigido al personal informático responsable de la instalación, en colaboración con los usuarios que van a usar la aplicación (clientes).

## Explotación:

La fase de explotación es la fase en la que los usuarios finales conocen la aplicación y comienzan a utilizarla.

Durante esta fase, se instalará la aplicación, se realizará la puesta a punto y se comprobará el funcionamiento de la aplicación en el equipo final del cliente. Los programas serán transferidos al computador del usuario cliente y se configurarán y verificarán. Haremos que los futuros clientes estén presentes en el proceso de instalación y le iremos comentando el proceso.

En este momento, llevaremos a cabo las Beta Test, que son las últimas pruebas que se realizan en los propios equipos del cliente y bajo cargas normales de trabajo.

Una vez instalada la aplicación, pasaremos a la fase de configuración. En ella, asignaremos los parámetros de funcionamiento normal y probaremos que la aplicación es operativa.

## Mantenimiento:

La etapa de mantenimiento es la más larga de todo el ciclo de vida del software. Éste es cambiante y deberá actualizarse y evolucionar con el tiempo. Deberá ir adaptándose de forma paralela a las mejoras del hardware en el mercado y afrontar situaciones nuevas que no existían cuando el software se construyó.

Por otro lado, siempre surgen errores que habrá que ir corrigiendo y nuevas versiones mejoradas del producto.

Por ello, se pactará con el cliente un servicio de mantenimiento de la aplicación. Este servicio, tendrá un coste temporal y económico.

Durante este proceso se realizará el control, mejora y optimización del software.

También tendremos en cuenta aspectos como nuevas necesidades del cliente, adaptaciones a nuevas tendencias del mercado o nuevos componentes software y la corrección de posible errores futuros.

Le ofreceremos al cliente un precio por un mantenimiento **perfectivo** (que mejorará su funcionalidad), **evolutivo** (modificaciones o expansiones para futuras nuevas necesidades), **adaptativo** (modificaciones o actualizaciones que se adapten a las nuevas tendencias del mercado o nuevos componentes hardware) y **correctivos** (para errores que pueda tener la aplicación en el futuro).