

# **ENTORNOS DE DESARROLLO**

**NOMBRE: Francisco Ortiz Tomás**

**CURSO: 1º DAM**

# TA04 Proceso de desarrollo de una aplicación

## Requisitos del trabajo:

### 1. Análisis de requisitos.

De manera general el cliente tiene varias necesidades que hay que cubrir o solventar por medio del desarrollo del software pertinente. Necesita una aplicación de software libre de licencias para una tienda de jardinería en la cual, de manera general se puedan controlar los siguientes aspectos.

- Generar facturas.
- Modificar facturas.
- Registrar las ventas.
- Control de stockage.
- Control del valor de los precios.
- Registro del personal que está efectuando una transacción.

Esta especificación de requisitos es completa , clara y concisa, ya que se realizará una función para cada requisito, también es consistente y se puede validar midiendo la eficacia de las funciones que van a realizar el software, por ejemplo es fácil identificar si los registros de productos o los registros de la base de datos se han guardado correctamente y son de fácil acceso. Estos requisitos se sintetizarán entre **Funcionales** y **No Funcionales**, y se codificarán para su seguimiento de forma modular.

### Funcionales:

**RF1.**- Generar facturas.

**RF2.**- Modificar facturas.

**RF3.**- Registrar las ventas realizadas por cada empleado.

**RF4.**- Control de stockage tras cada venta.

**RF5.**- Se deben recoger los datos con información relevante de los empleados.

**RF6.**- Se debe permitir la búsqueda de productos por nombre, marca o código.

### No Funcionales:

**RNF1.**- Actualización inmediata del stockage a través del uso de lectores de códigos de barra y tarjetas de crédito.

**RNF2.**- Los cambios de precios deben reflejarse inmediatamente en la app.

**RNF3.**- El tiempo de respuesta del sistema ha de ser en menos de 2 segundos.

**RNF4.-** No queremos que se procesen dos ventas del mismo artículo. La aplicación eliminará del stock el producto que se encuentre en transacción una vez se inicie la venta. Este sistema debe permitir que al escanear un producto, este se reste automáticamente del inventario disponible, asegurando que la venta solo se procese una vez y refleje el stock real.

**RNF5.-** Incluir un sistema de gestión de bases de datos donde se encuentren los datos de los empleados ( DNI, Nombre, Apellidos, N.<sup>º</sup> seguridad social,puesto que desempeña, antigüedad).

**RNF6.-** Incluir en el SGBD datos de los productos (código del producto, marca, nombre comercial, tipo de producto)

**RNF7.-** Registrar las ventas de los empleados para controlar las comisiones y el rendimiento individual.

**RNF8.-** El sistema debe ser **software libre**, sin costes por licencias.

**RNF9.-** El sistema debe ser multiplataforma (Linux y Windows al menos).

**2.Diseño.** Proporciona una propuesta de diseño del sistema, mencionando los módulos que interactuarán entre sí y seleccionando el **ciclo de vida de desarrollo o la metodología** que consideres más adecuada para este tipo de aplicación. Incluye al menos dos documentos de los que se generan en esta fase según los apuntes.

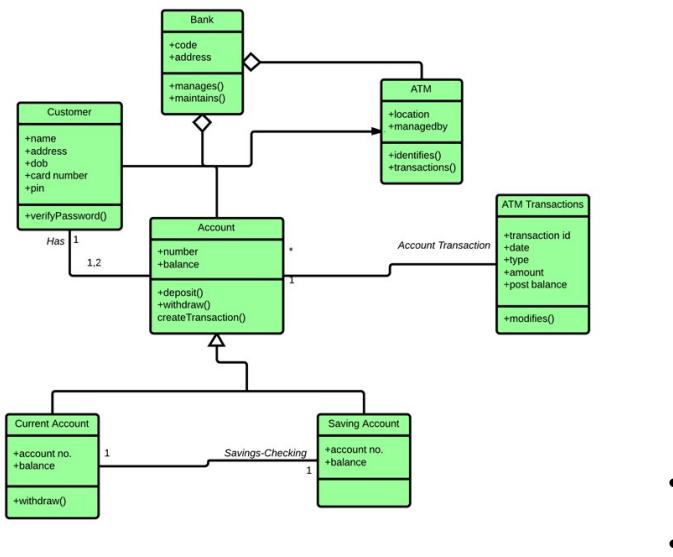
En esta fase se define cómo se va a construir el sistema que se ha descrito en la fase de análisis. No se escribe código todavía; se planifica y documenta la estructura técnica del software: sus módulos, sus relaciones, la base de datos y la interfaz de usuario.

### **Actividades principales**

- **Diseño arquitectónico:** se establece la estructura general del sistema y la relación entre sus módulos.
- **Diseño detallado:** se definen las funciones internas de cada módulo.
- **Diseño de datos:** se determinan las tablas, claves y relaciones de la base de datos.
- **Diseño de interfaz de usuario:** se define el aspecto visual y la interacción con el usuario.

## Documentos generados

**Diagrama de clases (POO):** muestra los objetos del sistema (Producto, Empleado, Venta, Cliente) con sus atributos y métodos.



**Diagrama entidad-relación (ER):** define las tablas de la base de datos y sus relaciones.

**Documento de arquitectura y módulos:** explica cómo se dividen las funcionalidades.

**Diseño de interfaz:** boceto de las pantallas de login, facturación y gestión de stock.

## Propuesta de diseño para la tienda de jardinería

### a) Módulos principales

Módulo	Funcionalidad principal
Gestión de Productos	Alta, baja, modificación y búsqueda de artículos.
Gestión de Ventas	Facturación, cobros, integración con TPV y lector de códigos de barras.
Gestión de Empleados	Registro de datos, control de ventas y comisiones.
Control de Stock	Actualización automática tras cada venta.

Módulo	Funcionalidad principal
Informes	Generación de listados de ventas, stock y rendimiento.

**3.Codificación.** Planifica el proceso de codificación, eligiendo el **lenguaje** de programación y las **herramientas** que utilizarás para obtener el código fuente, el código objeto y el ejecutable. Justifica la elección de estas herramientas.

#### a) Lenguaje y herramientas

- Lenguaje: **Java** (por su carácter libre y multiplataforma).
- Framework: En back-end **Django**.
- Base de datos: **MYSQL** (software libre, potente y seguro).
- Control de versiones: **Git / GitHub**.( debido a su versatilidad para trabajar con código en equipo)
- IDE recomendado: **Visual Studio Code**.

#### b) Proceso

1. Generar código fuente en eclipse (Java).
2. Compilar el código usando las librerías pertinentes a través del enlazado.
3. Generar ejecutable

#### c) Metodología de desarrollo

Se selecciona la **metodología incremental**, por ser más flexible y adecuada para proyectos pequeños y con interacción continua con el cliente.

Cada versión entregará una parte funcional (por ejemplo, primero facturación, luego stock, luego informes).

Esta metodología permite la **revisión continua** por parte del cliente.

Reduce los riesgos de errores de diseño tardíos (problema del modelo en cascada).

#### **4. Desarrollo de las fases restantes del ciclo de vida del software.**

Describe el objetivo de cada fase (pruebas, mantenimiento, ...) y explica cómo abordarías tú cada una de ellas en este proyecto.

<b>Fase</b>	<b>Objetivo</b>	<b>Aplicación al proyecto</b>
<b>Implementación</b>	Instalar el sistema en los equipos del cliente.	Aquí escribimos el código y crearemos el repositorio del mismo.
<b>Pruebas</b>	Detectar y corregir errores antes de la entrega.	Pruebas unitarias (Junit), de integración y de aceptación de sistema.
<b>Explotación</b>	El software es instalado en los equipos de la tienda y empieza a usarse para detectar fallos y errores	Los trabajadores recibirán un manual de usuario con instrucciones para hacer un correcto uso de la aplicación.
<b>Mantenimiento</b>	Corregir errores y realizar mejoras evolutivas.	Se hará un mantenimiento correctivo y evolutivo para corregir defectos y añadir nuevas funcionalidades.
<b>Documentación</b>	Facilitar el uso y comprensión del sistema.	De manera transversal iremos recogiendo la documentación que se generará en cada una de las fases de desarrollo. (Manual de usuario, diagramas de casos de uso, etc).
<b>Retirada</b>	Eliminar la huella de datos, y dejar de desarrollar el programa debido a la falta de rentabilidad por el coste de ampliación y mantenimiento.	Cuando el software llegue al fin de su vida útil se retirará y se eliminarán todos los datos que podrán ocasionar futuros problemas legales debido a la (ley de protección de datos).