

Sintetizar el análisis de requerimientos del sistema para el cliente. Plantear el diseño y determinar el modelo de ciclo de vida más idóneo para esta aplicación.

Se recibe una petición de la empresa SoftwareIng en la que contratan el desarrollo, preferiblemente sobre software libre, de una aplicación para una tienda de productos culinarios.

Requisitos funcionales:

- Plataforma de compraventa (funciones a realizar):
 - Generación de facturas de los pedidos realizados
 - Modificación del precio de los productos
 - Actualización de existencias tras venta
 - Reposición de productos con pocas existencias
 - Lectura de los códigos de barras de los productos
 - Procesamiento de pagos mediante tarjeta de crédito
- Productos (información para almacenar):
 - Código de producto
 - Marca
 - Nombre comercial
 - Cantidad (existencias, *stock*)
 - Precio
 - Código de barras
- Trabajadores (información para almacenar):
 - DNI
 - Nombre
 - Apellidos
 - Número de la Seguridad Social
 - Fecha de nacimiento
 - Teléfono
 - Localidad
 - Periodo de vacaciones
 - Ventas realizadas

Requisitos no funcionales:

- No se podrán efectuar peticiones de forma simultánea
- La aplicación deberá ejecutarse de forma liviana

En cuanto al diseño, optamos por dividir el proyecto en dos partes:

1. Gestión: incluye todas las funciones anteriormente listadas en “Plataforma de compraventa”, ya que serán las operaciones que permitirá realizar la aplicación.
2. Almacenamiento: especificamos qué datos necesitamos almacenar para el correcto funcionamiento de la aplicación.

En lo relativo al ciclo de vida del proyecto, se opta por elegir el modelo en cascada con realimentación, puesto que los requisitos son claros y están bien definidos; de todos modos, en caso de que la empresa quisiese modificar algún requisito, este modelo nos permitiría realizar esos cambios sin tener que empezar de cero todo el proceso.

Planificar la codificación, indicando el lenguaje de programación y las herramientas que se usarían para la obtención del código fuente, objeto y ejecutable, explicando por qué se han elegido éstas y no otras.

Para lograr lo anteriormente necesario, emplearemos:

- Base de datos: PostgreSQL
 - El motivo de elegir PostgreSQL es debido a que es considerado el SGBD de código abierto más avanzado; además, es accesible desde múltiples lenguajes de programación.
 - Como alternativa, se planteó el uso de MySQL, pero se optó por PostgreSQL al ser desarrollada de forma íntegra por la comunidad.
- Lenguaje de programación: Java
 - El motivo de elegir este lenguaje es porque permite ser ejecutado independientemente del hardware sobre el que esté instalado. Además, su «facilidad de uso», junto con su amplia cuota de mercado, permite desarrollar la aplicación de una forma rápida apoyándonos en la documentación existente.
 - El proceso para la obtención de código será el siguiente:
 - Se programará el código fuente en Java usando el Framework Spring.
 - Se compilará a código objeto (bytecode) gracias a las herramientas que proporciona el JDK.
 - Se transformará a código máquina gracias a la JVM.

Planificar las restantes fases del ciclo de vida, indicando en cada una el objetivo que se persigue y cómo se haría.

- Pruebas: comprobaremos que el programa funciona tal y como se espera mediante dos procedimientos previamente preparados en la fase de codificación:
 - Pruebas unitarias: se prueban las distintas partes de la aplicación, asegurándonos de que todas las funciones entregan los resultados esperados. Con dichos resultados generamos un documento de procedimiento de pruebas en el que se compararán los resultados obtenidos y los esperados.
 - Pruebas de integración: pasadas todas las pruebas unitarias, se comprobará que el programa, en conjunto, funciona como se esperaba; se generará un documento de procedimiento de pruebas de integración en el que también se compararán los resultados obtenidos y los esperados.
- Documentación: es una tarea que se realiza durante todo el ciclo de vida. Mediante la documentación dejaremos reflejada toda la información relativa al programa que hemos desarrollado. Aunque hay principalmente tres tipos, en este proyecto se harán dos:
 - Guía técnica: queda documentado cómo se ha diseñado y codificado la aplicación así como las pruebas realizadas. Nos permitirá realizar un mantenimiento eficaz.
 - Guía de uso: se documentará la aplicación desde el punto de vista del usuario final, con ejemplos de uso, requisitos de funcionamiento e incluso solución de problemas que se puedan presentar.

- Explotación (verificación en cliente): instalaremos y realizaremos una puesta a punto de la aplicación en las instalaciones de SoftwareIng.
 - Se realizará, primeramente, pruebas (Beta Test) en los equipos de la empresa para comprobar que todo funciona correctamente.
 - Una vez superados, se procederá a comprobar que los usuarios de la aplicación consiguen realizar todas las tareas esperadas de forma satisfactoria.
- Mantenimiento: implica todas las mejoras o correcciones a implementar en el programa entregado. Tendremos en cuenta cuatro tipo de cambios:
 - Perfectivos: mejorar ciertas funciones del programa.
 - Evolutivos: SoftwareIng solicita que el programa realice nuevas tareas. Empezaría de nuevo el proceso de análisis para añadir esas funciones. Por ejemplo, almacenar información de clientes.
 - Adaptativos: modificaciones para adaptar el programa a nuevos componentes o a cambios legislativos, por ejemplo.
 - Correctivos: corregir los errores (*bugs*) detectados durante el uso cotidiano de la aplicación.