

Tarea 1. Desarrollo de un proyecto

1. Explica el modelo de ciclo de vida del desarrollo de software que seguirás para este proyecto. Justifica la elección del ciclo de vida que mejor se adapta a este proyecto argumentando la decisión.
2. Análisis: Sintetiza el ERS clasificando los requisitos en funcionales y no funcionales.
3. Diseño:
 1. Identifica los datos que debe gestionar la aplicación.
 2. Selecciona lenguaje para desarrollo
 3. Selecciona herramienta para gestionar los datos
4. Codificación: Identifica las fases así como las herramientas a utilizar en cada una (edición de código, compilación, etc.)
5. Pruebas: Identifica el tipo de pruebas que se harán.
6. Explotación: Tareas a realizar.
7. Mantenimiento
8. Documentación: Relaciona los tipos de documentos que se van a generar y en qué partes del proyecto.

1. Mi pensamiento principal ha sido el modelo en cascada debido a que se prevén pocos cambios y es un proyecto pequeño. Pero reflexionándolo detenidamente he optado por el modelo de cascada con realimentación.

Al ser un proyecto para gestionar un inventario puede ser que se deba hacer a largo plazo alguna modificación o función nueva, por cambio de temporada, de materia, nuevos ingresos, actualización etc. Si bien, se hace hincapié en que no se esperan a corto plazo, no se descarta del todo.

2.

<u>Funcionales</u>	<u>No funcionales</u>
<ul style="list-style-type: none">- Acceso simultáneo.- Gestión de productos.- Debe generar un informe diario de entradas y salidas.- Inventario en tiempo real.- Registro de albaranes electrónicos.- Autenticación de usuarios.	<ul style="list-style-type: none">- Compatible para usarse en otros S.O.- Tiempo de respuesta que no supere los 2 segundos.- Debe ser seguro, respetando la normativa de protección de datos.

3.

a) Datos:

- Información de los productos que se van a tener en el inventario; cantidad, clasificación o categoría, proveedor, precio, descripción.
- Usuarios: para el control de personas que usaran la aplicación, como y cuando.
Tipo de usuario, nombre, contraseña, movimientos realizados y hora de ellos.
- Informes y albaranes: Recepciones de carga, proveedor y trabajador, tipo de movimientos, etc.

Entornos de desarrollo
Jenifer Sierra Tobias

b) El lenguaje que elegiría sería java, no tendríamos problema a la hora de la compatibilidad entre s.o., además tiene un buen rendimiento y seguridad, algo necesario para el manejo de datos simultáneos que nos pide la empresa.

c) PostgreSQL, es de código abierto y tiene gran compatibilidad, también cuenta con funciones de cifrado y seguridad.

4.

- a) Edición de código: Eclipse.
- b) Retroceso: Git.
- c) Compilación: Maven
- d) Enlace: JDK

5.

- Comprobar que cada módulo funciona de manera correcta, individual y conjuntamente.
- Verificar que el acceso simultáneo es correcto y no afecta al rendimiento.
- Asegurarse de que no puedan acceder más de 10 personas de manera simultánea ni personas no autorizadas.
- Probar el rendimiento y compatibilidad de la aplicación en diferentes S.O.
- Verificar que se generan el informe diario y se registra correctamente los movimientos solicitados.
- Simular el uso de albarán electrónico para comprobar su buen funcionamiento.

6.

- Instalación en los equipos correspondientes.
- Realización de las betas test.
- Configuración del sistema.

7. Resolvemos los errores que el cliente nos haya podido comunicar e implementamos las actualizaciones correspondientes.

8.

- Durante el análisis: documento que detalle los requisitos del sistema y todos los detalles a tener en cuenta.
- Diseño: Documento de diseño del software y plan de pruebas.
- Codificación: Pautas de codificación, lógica de los movimientos y registro de cambios.
- Pruebas: Resultado y uso del plan de pruebas.
- Fase final: Guía técnica, guía de uso y guía de instalación.

Tarea 2. Modelos de desarrollo ágil

1. Explica el principal problema de los modelos clásicos y qué alternativa proponen los modelos de desarrollo ágil.
2. ¿Para qué tipos de proyectos son más indicados este tipo de modelos? Razona la respuesta.
3. Explica brevemente las similitudes y diferencias de la Programación Extrema (XP) y SCRUM.

1. Los modelos clásicos son más rígidos y menos evolutivos. Tienen una serie de fases necesarias seguir de manera estricta y documentada. También la comunicación o colaboración del cliente se ve reducida, ya que no ve el proyecto hasta su fase final, lo que puede dificultar quedar satisfecho con el resultado.

Los modelos de desarrollo ágil se centran en el presente del proyecto y la satisfacción del cliente. Dividen el proceso en pequeños sprints, los cuales hacen que el cliente este más integrado , ya que, tiene mas oportunidades para evaluarle.

2. Para proyectos con requisitos pocos claros o que necesites cambios constantes.

3.

- Similitudes: Ambos buscan la interacción constante del cliente y se basan en pequeños ciclos de producción.

- Diferencias: SCRUM es un poco más “organizado”, define desde un principio los roles y requisitos de participante y fase, en cambio XP, es más flexible y se enfoca más en la colaboración del equipo.