CAPÍTULO 1

COMPRUEBA TU APRENDIZAJE

1) ¿Cuál de las siguientes afirmaciones es correcta?

a. En los modelos evolutivos no se necesita conocer todos los requisitos al comienzo.

b. Es muy común en el modelo en cascada el uso de prototipos.

c. El análisis de riesgos se lleva a cabo en cada incremento del modelo iterativo incremental.

d. El modelo en cascada es apropiado cuando se necesita una versión inicial del software a desarrollar.

2) Relaciona:

Modelo en cascada (1) Modelo iterativo incremental (2) Modelo en espiral (3)	Es fácil de comprender. (2) Los clientes necesitan versiones intermedias. (1) No se necesita conocer todos los requisitos al comienzo. (3) Reduce riesgos del proyecto. (2) Los requisitos son estables. (1) Genera mucho trabajo adicional. (3) No se sabe cuando va a terminar. (3) Los requisitos son estables. (1) El proyecto es similar a uno ya realizado (1) Se acomoda bien a los cambios de requisitos. (2)

3) ¿Qué se hace en la etapa de análisis del desarrollo de una aplicación?

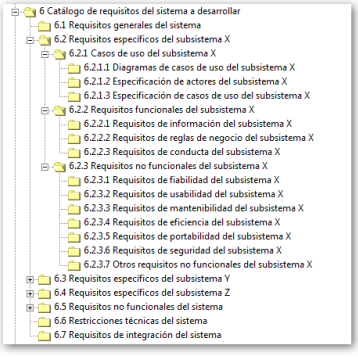
Se especifican y analizan los requisitos funcionales y no funcionales del sistema, siendo lo fundamental una buena comunicación entre el analista y el cliente para que la aplicación que se va a desarrollar cumpla con las expectativas del cliente.

¿Qué tipos de requisitos se especifican en esta fase? (Nombra algunas herramientas para obtener y representar los requisitos)

- Requisitos Funcionales
  - ¿Qué funciones…
  - ¿Qué respuestas…
  - ¿Qué comportamiento…

  … tendrá la aplicación?
- Requisitos No Funcionales
  - Tiempo de respuesta…
  - Legislación aplicable…
  - Simultaneidad de peticiones…

  … del programa



Una herramienta muy buena para llevar a cabo y culminar esta etapa de desarrollo, es el documento ERS, también conocido como *Especificación de Requisitos de Software*

4) ¿Cuál de estas afirmaciones sobre la fase de diseño en el desarrollo de una aplicación es correcta?

a. En esta fase se especifica qué hay que hacer.

b. En esta fase se especifica cómo hacerlo.

c. En esta fase se realiza el proceso de programación.

d. En esta fase se realizan las pruebas.

5) ¿En qué consisten las tareas de verificación y validación del software?

Consisten en realizar pruebas sobre el software

6) ¿Qué tareas se llevan a cabo durante la explotación del sistema?

- Instalación
- Puesta a punto
- Funcionamiento de la aplicación en el equipo final del cliente
- Beta Test
- Configuración
- Producción normal

7) ¿Cuántos tipos de mantenimiento del software existen? ¿En qué consisten?

i. Perfectivos: Para mejorar la funcionalidad del software.

ii. Evolutivos: El cliente tendrá en el futuro nuevas necesidades. Por tanto, serán necesarias modificaciones, expansiones o eliminaciones de código.

iii. Adaptativos: Modificaciones, actualizaciones... para adaptarse a las nuevas tendencias del mercado, a nuevos componentes hardware, etc.

iv. Correctivos: La aplicación tendrá errores en el futuro (sería utópico pensar lo contrario).

8) ¿Qué tareas se llevan a cabo en la etapa de mantenimiento del software?

9) ¿Qué relación tiene un programa con el hardware del ordenador donde se ejecuta?

La principal diferencia entre hardware y software es que el hardware es todo dispositivo físico, algo que se puede tocar, al contrario del software que es un conjunto de instrucciones de código instalado en el computador que se ejecutan para cumplir una función, no lo puedes tocar físicamente.

10) ¿De qué tres elementos consta un lenguaje de programación?

- Alfabeto: conjunto de símbolos permitidos.
- Sintaxis: normas de construcción permitidas de los símbolos del lenguaje.
- Semántica: significado de las construcciones para hacer acciones válidas.

11) ¿Qué diferencia hay entre un lenguaje de alto nivel y otro de bajo nivel?

Que los Lenguajes de Alto Nivel están más próximos al razonamiento humano, mientras que los Lenguajes de Bajo Nivel están más próximos al razonamiento de una máquina (funcionamiento interno del ordenador).

12) Diferencias entre un compilador y un intérprete.

La principal diferencia reside en que un compilador pasa el código fuente de programación a código objeto; y por otro lado, el intérprete traduce directamente el código fuente al lenguaje máquina.

13) ¿Qué ventajas aporta la programación modular?

Permite dividir los programas grandes en trozos más pequeños (bloques) permitiendo mayor funcionalidad