

# TeoriaTema2ElProcesodelSoftware.pdf



yavae



Ingeniería del Software



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática  
Universidad Politécnica de Valencia

## Encuentra **el trabajo** de tus sueños

### Participa en retos y competiciones de programación

Ten contacto de calidad con empresas líderes en el sector tecnológico mientras vives una experiencia divertida y enriquecedora durante el proceso.

**Únete ahora**



Escanéame y  
obtén más info!!



# Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación



Escanéame y  
obtén más info!!

## TeoríaT2 El Proceso del Software

Yavae

### 1. Comenta la siguiente afirmación: “La Ingeniería del Software es algo más que programar...”

La Ingeniería del Software es una disciplina que se encarga del desarrollo, mantenimiento y evolución de software, abarcando desde la planificación hasta la evaluación y mejora continua del mismo. Incluye aspectos como la gestión de proyectos, la especificación de requisitos, el diseño, la construcción, la prueba y la documentación del software.

### 2. ¿Qué causas motivaron la necesidad de aplicar un enfoque ingenieril al desarrollo de software?

La necesidad de aplicar un enfoque ingenieril al desarrollo de software surgió debido a la complejidad y el alcance creciente del software en las organizaciones, así como también la necesidad de mejorar la calidad del software y reducir los costos y riesgos asociados al desarrollo de software.

### 3. Define el término “Proceso del Software”. Define el término “Modelo de Proceso Software”.

El proceso de software se refiere al conjunto de actividades y tareas necesarias para desarrollar, mantener y evolucionar un software. El modelo de proceso software es un marco que describe cómo se deben llevar a cabo estas actividades y tareas en un proyecto específico.

### 4. ¿Existe algún modelo de proceso que incorpore prototipado? ¿Cuál? ¿Cuál es su objetivo?

El modelo de proceso que incorpora prototipado es el modelo clásico con prototipado, su objetivo es obtener retroalimentación temprana de los usuarios y adaptarse mejor a los cambios en los requisitos.

### 5. ¿Qué significa que un proceso de desarrollo de software sea iterativo, evolutivo e interactivo?

Un proceso de desarrollo de software iterativo, evolutivo e interactivo se refiere a un proceso en el cual el software se desarrolla a través de ciclos de iteraciones, donde cada iteración se basa en la retroalimentación recibida en la iteración anterior. El proceso es evolutivo ya que el software se desarrolla a través de versiones incrementales y se adapta a los cambios en los requisitos. El proceso es interactivo ya que se fomenta la colaboración entre los miembros del equipo y los usuarios.

### 6. ¿Qué inconvenientes presenta el desarrollo en cascada? ¿Cuándo sería apropiado utilizarlo?

El desarrollo en cascada presenta inconvenientes como la falta de flexibilidad ante cambios en los requisitos, la dificultad para identificar y abordar los riesgos de manera temprana, y la falta de retroalimentación continua de los usuarios. Sin embargo, puede ser apropiado utilizarlo en proyectos con requisitos bien definidos y estables, donde no se esperan cambios significativos en los requisitos.

### 7. Explica cuatro características relevantes del paradigma de programación automática.

El paradigma de programación automática se caracteriza por la utilización de algoritmos y técnicas automatizadas para el desarrollo de software. Algunas de sus características relevantes son:

- La utilización de lenguajes de programación formales y notaciones matemáticas para especificar el comportamiento del software.
- La utilización de herramientas y técnicas automatizadas para la verificación y validación del software.
- La utilización de técnicas de optimización automática para mejorar el rendimiento y la eficiencia del software.
- La utilización de técnicas de inteligencia artificial para mejorar la toma de decisiones en el desarrollo del software.

### 8. Explica qué diferencias existen entre el ciclo de vida incremental y el espiral.

El ciclo de vida incremental se caracteriza por desarrollar el software en etapas o versiones incrementales, donde cada versión es un subconjunto completo y usable del software final. El ciclo de vida espiral se caracteriza por evaluar y mitigar los riesgos en cada iteración del proceso de desarrollo, mientras se desarrolla el software. Ambos modelos buscan mejorar la retroalimentación y adaptabilidad a los cambios en los requisitos.



**9. ¿Qué diferencia existe entre los siguientes factores de calidad del software: corrección, fiabilidad e integridad?**

El factor de calidad corrección se refiere a la capacidad del software de cumplir con sus especificaciones y funcionar correctamente. La fiabilidad se refiere a la capacidad del software de funcionar de manera consistente y predecible en condiciones normales de operación. La integridad se refiere a la capacidad del software de proteger y preservar los datos y recursos del sistema.

**10. Explica por qué el proceso de redacción del plan de proyecto es iterativo y por qué debe revisarse continuamente durante el proyecto.**

El proceso de redacción del plan de proyecto es iterativo ya que a medida que se avanza en el proyecto, se obtiene más información y se pueden identificar nuevos requisitos o cambios en los existentes. Por esta razón, es importante revisar continuamente el plan de proyecto para asegurar que sigue siendo relevante y estratégico para el proyecto.

**11. En la Administración de un proyecto software, ¿qué es un hito? Pon un ejemplo.**

En la Administración de un proyecto software, un hito es un evento o logro importante que marca el progreso del proyecto. Ejemplo: la finalización de una fase de desarrollo o la entrega de una versión beta al cliente.

**12. Identifica cuatro posibles riesgos que pueden surgir en un proyecto software. Explica a qué afectarían o podrían afectar en el proyecto.**

Posibles riesgos en un proyecto software son: 1) Cambios en los requisitos del cliente, lo que afectaría al alcance y al cronograma del proyecto. 2) Problemas con los recursos, como la falta de habilidades técnicas o una falta de tiempo, lo que afectaría al desempeño del equipo. 3) Problemas con el presupuesto, lo que afectaría al costo y al margen de beneficio del proyecto. 4) Problemas con la tecnología, como la dependencia de una herramienta o un sistema que no está disponible, lo que afectaría a la calidad y la fiabilidad del producto final.

**13. Explica al menos cuatro causas de la siguiente situación: “Los productos software no son de calidad, sus costes sean elevados y su entrega sufre retrasos importantes”**

Algunas causas de una situación en la que los productos software no son de calidad, sus costes son elevados y su entrega sufre retrasos importantes son: 1) Falta de una planificación adecuada y un seguimiento riguroso del proyecto. 2) Cambios en los requisitos del cliente no gestionados adecuadamente. 3) Falta de habilidades técnicas en el equipo de desarrollo. 4) Problemas con la infraestructura o las herramientas de desarrollo.

**14. ¿Qué se entiende por el término “software de calidad”?**

El término "software de calidad" se refiere al grado en el que un software cumple con los requisitos y es fiable, usable y eficiente. Esto incluye aspectos como la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la portabilidad. La calidad del software es esencial para garantizar que el software cumple con las necesidades del usuario y es fácil de usar e mantener.