

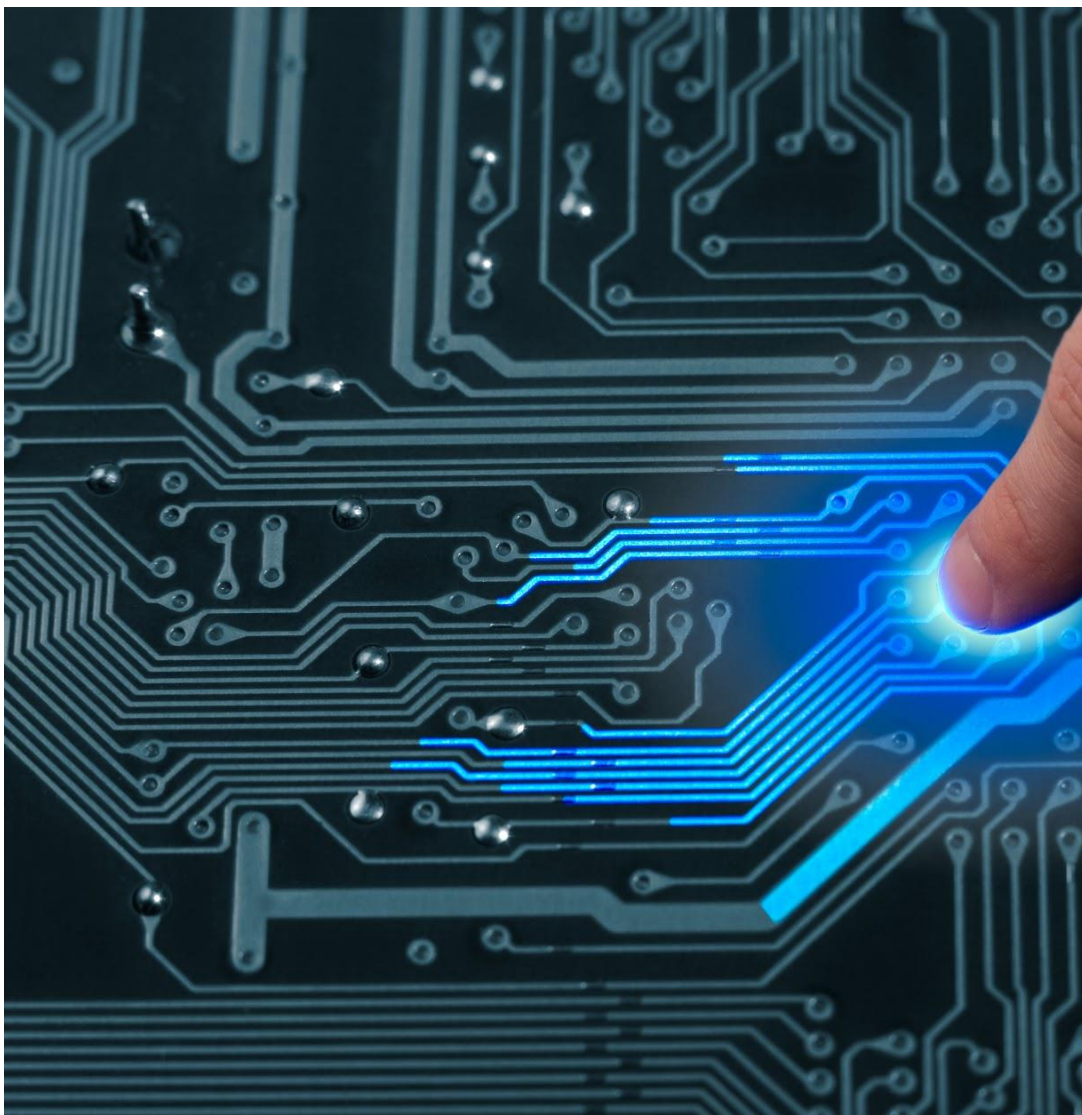
CFGs

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Entornos de desarrollo

DESARROLLO DE SOFTWARE

PAC 1: Desarrollo de software



PAC 1: Desarrollo de software

INFORMACIÓN

Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y consultar internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del libro digital para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Se valorará la presentación, ortografía y gramática de vuestro trabajo hasta con un punto de la nota final.

1. ¿Encuentras alguna diferencia entre el freeware y el software libre? Explícalas.

Aunque ambos tipos de software se distribuyen sin ningún coste, la principal diferencia es que su clasificación se realiza por diferente motivo. El software libre según el tipo de licencia, y el freeware según el método de distribución. Esta diferencia se refleja en que la distribución del freeware no incluye el código fuente, y en caso de que lo incluya, no se permite su modificación, ni su venta. Mientras que, el software libre permite su modificación, mejora y redistribución. Por lo que no todo freeware es software libre.

2. Explica con tus palabras el funcionamiento de un compilador.

Un compilador es el traductor del código fuente en código objeto, es decir, traduce el lenguaje de alto nivel a un medio nivel. El compilador es el encargado también de detectar los errores en este código fuente. Si el código fuente contiene errores, entonces el compilador sólo indicará la existencia de los mismos, y no creará el código objeto.

Para todo esto, el compilador analiza el código fuente de 6 maneras distintas:

- **Análisis léxico: se encarga de identificar los lexemas, es decir, identifica las palabras reservadas del lenguaje.**

- **Análisis sintáctico:** es encarga de comprobar si se cumplen las diferentes reglas sintácticas.
- **Análisis semántico:** se encarga de comprobar que las declaraciones en el código fuente son correctas.
- **Generación de código intermedio:** se analiza el código al completo para facilitar la traducción a código objeto.
- **Optimización de código:** se mejora el código generado anteriormente.
- **Generación de código:** se crea el código objeto.

3. Explica las ventajas y desventajas de la máquina virtual de java.

Ventajas:

- **Portabilidad.** Los lenguajes de programación ejecutados en una máquina virtual son independientes de la plataforma, por lo que podemos ejecutar estos programas en cualquier sistema operativo. Permite realizar aplicaciones multiplataforma.

Desventajas:

- **Son más lentos que los lenguajes compilados.**

4. ¿Cuáles son los elementos de los lenguajes de programación?

Consta de tres elementos; alfabeto, sintaxis y semántica.

- **El alfabeto** es el conjunto de símbolos permitidos.
- **La sintaxis** son las reglas para realizar las construcciones con los símbolos.
- **La semántica** son las reglas que determinan el significado de las construcciones realizadas.

5. Realiza la clasificación de los siguientes lenguajes: C, Java, C#, JavaScript, PHP, Prolog.

	Según la abstracción	Según la ejecución	Según la programación
C	Nivel medio	Compilado	Imperativo + funcional + estructurada
Java	Nivel alto	Compilado e interpretado	Imperativo + orientado a objetos
C#	Nivel alto	Compilado	Estructurado + Imperativo + funcional + orientado a objetos
JavaScript	Nivel alto	Interpretado	Imperativo + funcional
PHP	Nivel alto	Interpretado	Imperativo + funcional + orientado a objetos
Prolog	Nivel alto	Interpretado	Lógico

6. ¿Qué es un intérprete?

Programa con la misma funcionalidad que un compilador, con la diferencia del procesamiento que realiza en el código fuente, pero la traducción se realiza instrucción a instrucción, según lo va leyendo.

7. Realiza un resumen/esquema de las fases del desarrollo de software.

El ciclo de vida del software se compone de una serie de fases, tratando cada uno de los aspectos del desarrollo de una aplicación.

Fase 1: Análisis

Es la tarea más importante del proceso, puesto que en esta fase se recopilan los requisitos necesarios que la aplicación debe solucionar. Para la obtención de estos requisitos podemos emplear varias técnicas: entrevistas, prototipos, brainstorming, casos de uso..

Fase 2: Diseño

Es la fase en la que se traducen los requisitos recopilados durante el análisis. Se piensa el modo en que se llevará a cabo la resolución del problema, por ejemplo, los diagramas de flujo, de cajas, tablas de decisión o el pseudocódigo.

Fase 3: Codificación

En esta fase se transforman los requisitos en el lenguaje de programación anterior. Es conveniente seguir algunas pautas para un mejor mantenimiento posteriormente, como es el uso de comentarios, la nomenclatura de los diferentes elementos de programación y la organización de los ficheros que forman parte del proyecto.

Fase 4: Pruebas

Es la fase en la que se detectan los errores, tanto de la parte de codificación como en la especificación del diseño.

Fase 5: Documentación

Durante el desarrollo es necesario llevar a cabo el registro de toda la información necesaria del proceso y del producto. También es necesario realizar el manual de usuario, que explica cómo utilizar este producto.

Fase 6: Explotación

En esta fase es necesario llevar a cabo la instalación de la aplicación creada, de esta forma, el cliente ya puede utilizar el producto.

Fase 7: Mantenimiento

Durante esta fase se lleva a cabo la modificación del producto para corregir los fallos. Se deben tener en cuenta los distintos tipos de mantenimiento que existen; adaptativo, correctivo, perfectivo y preventivo.

8. Las fases de desarrollo siguen diferentes modelos de ciclo de vida. Investiga por internet sobre los modelos en cascada y evolutivos.

Modelo en cascada: las etapas del ciclo de vida siguen un orden, de tal forma que para cambiar de fase se tiene que haber finalizado la anterior.

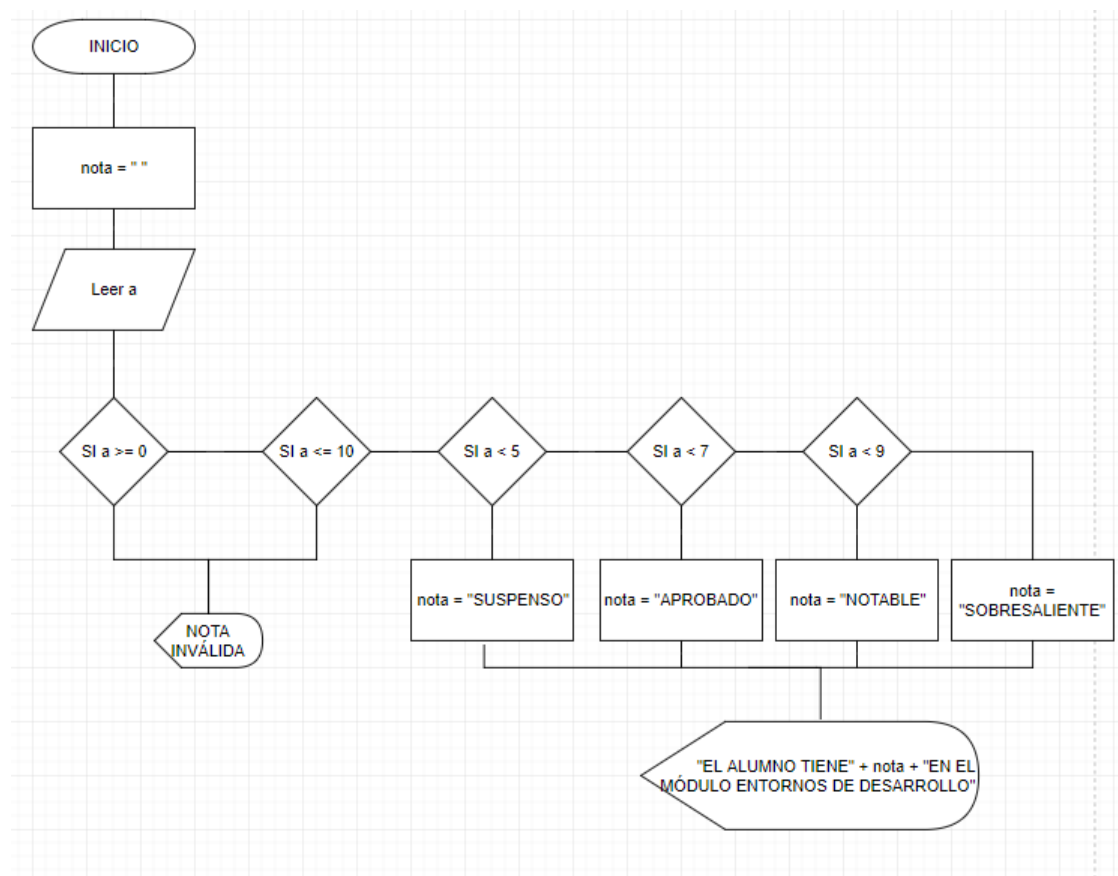
Modelos evolutivos: se van a ir realizando iteraciones sobre el ciclo de vida del software, permitiendo entregar versiones en cada una de estas iteraciones, mientras se van añadiendo mejoras.

Tipos:

- **Modelo iterativo incremental:** se compone de varios ciclos de vida en cascada realimentados, en el que cada vez que se realiza el ciclo completo se crea un incremento.
- **Modelo en espiral:** se combina el modelo en cascada con el modelo iterativo. Cada ciclo de la espiral está formado por 4 fases (determinar objetivos, identificar y resolver riesgos, desarrollar y probar, planificación), y cuando se termina el ciclo, se produce una versión como en el modelo iterativo.

9. Realiza el pseudocódigo y el diagrama de flujo para que, dada una nota entre 0 y 10, el programa muestre su nota en modo texto.

- Entre 0 y 4.99 -> SUSPENSO
- Entre 5 y 6.99 -> APROBADO
- Entre 7 y 8.99 -> NOTABLE
- Entre 9 y 10 -> SOBRESALIENTE



INICIO

ESCRIBE LA NOTA NUMÉRICA

LEER NOTA

SI $a \leq 0$ ENTONCES

MOSTRAR "NOTA INVÁLIDA"

SI NO

SI $a < 5$ ENTONCES

nota = "SUSPENSO"

SI NO

SI $a < 7$ ENTONCES

nota = "APROBADO"

SINO

SI $a < 9$ ENTONCES

nota = "NOTABLE"

SI NO

nota = "SOBRESALIENTE"

FIN SI

FIN SI

FIN SI

FIN SI

FIN

¡Buen trabajo!

