

Ciclo: Desarrollo de Aplicaciones Multiplataforma (DAM)

Módulo: Entornos de Desarrollo

Grupo: DAM1 (turno de mañana)

Examen tema 1 y 2.

Martes 10-11-2020

Desarrollo de Software, Entornos Integrados de Desarrollo.

Nombre y apellidos: Oliver Linares Colino

**Cuestión 1** (5 puntos): Lee atentamente las siguientes 20 preguntas, se muestran 3 respuestas de las cuales, deberás escoger la respuesta correcta o la más correcta, ten cuidado al elegir la respuesta dado que cada respuesta incorrecta resta puntos (cada respuesta acertada suma 0,25 puntos a la nota final del examen y cada respuesta errónea resta 0,08 puntos) (el valor mínimo de esta cuestión será de 0 puntos, es decir, no existen puntuaciones negativas).

1. De las siguientes definiciones de **Software** cuál consideras la más correcta:
  - a. Como el conjunto de procedimientos, cómputos, reglas y datos asociados que permiten a un sistema operativo apoyado un determinado hardware resolver ciertas tareas de un sistema computacional.
  - b. Como el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.
  - c. Como aquel conjunto de procedimientos y cómputos que permiten utilizar un determinado hardware computacional para, con ello, poder resolver ciertos problemas o tareas.

X →
2. Se denomina software **Freeware** a aquel software que:
  - a. Permite su modificación, distribución y utilización de forma libre y gratuita.
  - b. Se puede distribuir y utilizar de forma gratuita dado que carece de licencia o, bien, esta ha caducado o ha sido liberada.
  - c. Se distribuye sin cargo, pero no exento de cumplimiento de ciertas restricciones ligadas a derechos de copyright y licencias de uso.

✓
3. Una diferencia entre las licencias de **Software Libre** y **Open Source** es que:
  - a. La primera se centra en los detalles éticos de la compartición del conocimiento y la segunda en las ventajas técnicas de la compartición del código fuente.
  - b. La primera es un tipo de licencia que busca distinguirse de las licencias propietarias y la segunda, de hecho, no es un tipo de licencia, sino un movimiento que lucha por la liberación del código propietario.
  - c. La primera se trata de una licencia que permite compartir y distribuir el software, normalmente sin su código fuente que puede permanecer oculto al usuario, mientras que la segunda se trata de una licencia que se basa en compartir y distribuir el código fuente de las aplicaciones, para que cualquiera pueda usarlo, mejorarlo y compartirlo.

X →
4. Una definición de **ciclo de vida de software** es “marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de requisitos hasta la finalización de su uso”.
  - a. Esta definición se incluye en el estándar IEEE/ISO 12207-01
  - b. Esta definición se incluye en el estándar ISO/IEC 12207-01
  - c. Esta definición se incluye en el estándar IEC/IEEE 12207-1

X →

14 OK  
5 NOK  
3,1

- ✓ 5. Se puede realizar una división en **etapas del ciclo de vida del software** según:
- Una de las etapas es el mantenimiento, que se centra en mantener “vivo” el software, liberando nuevas versiones que lo hagan más atractivo por incorporar mejoras estéticas y de calidad o bien por incorporar nuevas funcionalidades.
  - Una de las etapas es el mantenimiento, que se centra en solucionar los defectos que puedan localizarse en el software, bien en la etapa de pruebas, bien en el uso del mismo por el cliente.
  - Una de las etapas es el mantenimiento, que puede ser distinguido en varios tipos, en particular, se distinguen dos de esos tipos: El mantenimiento adaptativo y el perfectivo. De modo que mientras el primero busca adaptar el software a los cambios de hardware y software en el entorno de ejecución, el segundo busca adaptar el software a las necesidades, o a la evolución de estas que puedan surgir en el entorno cliente.
- ✓ 6. Una de las ventajas del **modelo iterativo incremental** en el desarrollo de software es que:
- Se obtiene fácilmente una estimación del coste final en las primeras etapas.
  - Aunque la entrega del software se demora, la calidad de este es elevado.
  - No necesita el conocimiento de todos los requisitos desde el comienzo.
7. Durante el **análisis** de un problema destinado a un **desarrollo de software**.
- Son fundamentales las reuniones con el cliente para la elaboración de aquellos requisitos funcionales de la aplicación.
  - Son fundamentales las reuniones con el cliente para la elaboración de aquellos requisitos no funcionales de la aplicación.
  - Son fundamentales las reuniones con el cliente para la elaboración de aquellos requisitos funcionales y no funcionales de la aplicación.
8. Una **estructura procedimental** del tipo “MIENTRAS se cumpla la CONDICIÓN 1, HACER la SECUENCIA 1” es una estructura:
- Secuencial.
  - Condicional.
  - Repetitiva.
- ✗ 9. Cuando compilamos una aplicación con extensión **.java**, el compilador nos devuelve un fichero con extensión **.class**. Se trata de un fichero de tipo:
- Código fuente.
  - Código objeto.
  - Código ejecutable.
- ✓ 10. En un proceso de desarrollo de software, el **informe de pruebas**:
- Se trata de un informe de las pruebas que será necesario realizar según cada caso de pruebas previsto.
  - Es un informe de los errores hallados en las pruebas, así como la depuración de estos como resultado del proceso.
  - Muestra la evaluación de los resultados obtenidos para cada caso de pruebas en función de los esperados.
- ✓ 11. En un **diagrama de flujo**, un rectángulo representa:
- Los procesos repetitivos o incluidos dentro de bucles.
  - Los procesos de entrada y salida de datos.
  - Los procesos o cálculos internos.
- ✓ 12. El uso de **pseudocódigo** es ventajoso porque:
- Permite realizar la programación de forma visual y sin necesidad de conocer el lenguaje de programación.
  - Permite ilustrar un programa o algoritmo centrándose en las estructuras y procedimientos, pudiendo así ser portado con facilidad a cualquier lenguaje de tipo procedimental.
  - Es lo que nos permite el paso de los diagramas de flujo al código fuente del lenguaje de programación utilizado.

Ciclo: Desarrollo de Aplicaciones Multiplataforma (DAM)

Módulo: Entornos de Desarrollo

Grupo: DAM1 (turno de mañana)

13. En cuanto a los lenguajes de programación:

- a. Un lenguaje de bajo nivel es aquel más cercano al lenguaje utilizado por la máquina, pero que precisamente por ese motivo sólo puede ser usado para desarrollos sencillos debido a la falta de recursos de programación que expone.
- b. Un lenguaje de alto nivel es aquel de gran complejidad intrínseca y que, por tanto, requiere mayor esfuerzo por el programador, así como que siempre expone una curva de aprendizaje mayor.
- c. Un lenguaje interpretado, no requiere ser compilado antes de su ejecución, sino que se realiza de forma dinámica en la misma.

14. En la ejecución de un programa según la arquitectura de Von Neumann, se pueden distinguir varias fases:

- a. La fase de búsqueda y la fase de ejecución.
- b. La fase de búsqueda, la fase de lectura, la fase de ejecución y la fase de escritura.
- c. La fase de búsqueda, la fase de ejecución y la fase de lectura/escritura.

15. Si escribimos un programa JAVA en un IDE, pero al introducir el código cometemos un error sintáctico, la herramienta que nos ayudará a encontrar dicho error será:

- a. El compilador.
- b. El depurador de código.
- c. El intérprete.

*Editor*

16. Si escribimos un programa JAVA en un IDE, pero uno de los algoritmos es incorrecto y los resultados que proporciona el programa no son correctos, la herramienta que mejor nos puede asistir en la búsqueda del error será:

- a. El editor.
- b. El depurador de código
- c. El intérprete.

17. Para poder compilar un programa en JAVA, necesitaremos las herramientas proporcionadas por el software:

- a. JAVA Runtime Environment (JRE).
- b. JAVA Developer's Kit (JDK).
- c. Ambas.

18. Cuál es el ejecutable que es necesario invocar, de los presentes en el Kit de Desarrollo JAVA, para compilar un programa.

- a. compiler.exe
- b. java.exe
- c. javac.exe

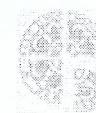
19. El plugin ERMaster de Eclipse nos permite:

- a. Crear aplicaciones con uso de ventanas mediante un entorno visual.
- b. Crear la estructura de una base de datos a partir de su diagrama Entidad-Relación.
- c. Crear la estructura de una base de datos a partir de su diagrama relacional.

20. Una diferencia importante entre Eclipse y NetBeans es que Eclipse permite programar en distintos lenguajes, mientras que NetBeans sólo soporta JAVA.

- a. Es incorrecto, en la actualidad NetBeans también es un IDE multilenguaje.
- b. Es incorrecto, aunque NetBeans está pensado para programar en JAVA, mediante plugins puede soportar otros lenguajes de programación como el C++.
- c. Es incorrecto, NetBeans es un IDE multilenguaje, sin embargo, Eclipse es un IDE que sólo soporta JAVA como lenguaje de programación.





Ciclo: Desarrollo de Aplicaciones Multiplataforma (DAM)

Módulo: Entornos de Desarrollo

Grupo: DAM1 (turno de mañana)

**Cuestión 4** (1 punto): Realiza en pseudocódigo un programa que lea dos números A y B, los multiplique y si el resultado P de dicho producto es mayor que la suma S de los primeros A números, entonces devuelva la resta de P menos S, en caso contrario devuelva la suma de P más S.

OBSERVACIÓN: para calcular la suma de los primeros A números, **no utilices el término matemático generador de la serie**, utiliza un bucle definido por el valor A (no se trata de un ejercicio de matemáticas sino de técnicas de programación).

INICIO

Leer A

Leer B

 $P = A * B$  $S = 0$ Para  $i = 1$  hasta  $A+1$  con Paso 1 Hacer $S = S + i$ 

FinPara

Si  $P > S$  EntoncesMostrar  $P - S$ 

Si No

Mostrar  $P + S$ 

FinSi

FIN

OK

Cuestión 5 (1 punto): Dibuja el diagrama de flujo del siguiente pseudocódigo:

```
INICIO
    Leer N
    SumaPar = 0
    ProductoImpar = 1
REPETIR
    Leer NumeroLista
    SI ((NumeroLista MOD 2) = 1) ENTONCES
        ProductoImpar = ProductoImpar * NumeroLista
    SI NO
        SumaPar = SumaPar + NumeroLista
    FIN SI
    N = N - 1
HASTA QUE (N = 0)
Escribir "La suma de los números Pares es:" SumaPar
Escribir "El producto de los números Impares es:" ProductoImpar
FIN
```

