

1. Para fabricar un programa informático que se ejecuta en una computadora:

- a) Hay que escribir las instrucciones en código binario para que las entienda el hardware.
- b) Sólo es necesario escribir el programa en algún lenguaje de programación y se ejecuta directamente.
- c) Hay que escribir el programa en algún Lenguaje de Programación y contar con herramientas software que lo traduzcan a código binario.
- d) Los programas informáticos no se pueden escribir: forman parte de los sistemas operativos.

2. Según estimaciones, el 26% de los grandes proyectos de software fracasan, el 48% deben modificarse drásticamente y sólo el 26% tienen rotundo éxito. La principal causa del fracaso de un proyecto es la falta de una buena planificación de las etapas y mala gestión de los pasos a seguir. ¿Por qué el porcentaje de fracaso es tan grande? ¿Por qué piensas que estas causas son tan determinantes?

3. Si queremos construir una aplicación pequeña, y se prevé que no sufrirá grandes cambios durante su vida, ¿sería el modelo de ciclo de vida en espiral el más recomendable?

- a) Sí
- b) No

4. Para obtener código fuente a partir de toda la información necesaria del problema:

- a) Se elige el Lenguaje de Programación más adecuado y se codifica correctamente.
- b) Se codifica y después se elige el Lenguaje de Programación más adecuado.
- c) Se elige el Lenguaje de Programación más adecuado, se diseña un algoritmo y se codifica.

5. Detecta el lenguaje según el fragmento de código indicado e indica si se trata de un lenguaje estructurado, orientado a objetos, de bajo nivel, de alto nivel, de nivel medio (este tipo de lenguajes tienen ciertas características que los acerca a los lenguajes de bajo nivel, pero a la vez también tienen características de los lenguajes de alto nivel), compilado, interpretado o compilado e interpretado al mismo tiempo. Para ello, debes saber:

Dependiendo de la forma en la que son ejecutados, puede hacerse otra clasificación:

- a) *Lenguajes compilados*. Necesitan un programa traductor (compilador) para convertir el código fuente en código máquina. Este tipo de programas se ejecutan de forma más rápida que los interpretados o los virtuales. Además del compilador, existe un programa llamado *enlazador* o *linker* que permite unir el código objeto del programa con el código objeto de las librerías.
- b) *Lenguajes interpretados*. No se genera código objeto. El intérprete es un programa que tiene que estar cargado en memoria y se encarga de leer cada una de las instrucciones, interpretarlas y ejecutarlas. Las instrucciones se traducen *on the fly* y solamente aquellas que van ejecutándose, en vez de interpretar todo el programa.
- c) *Lenguajes virtuales*. Son lenguajes más portables que los lenguajes compilados, puesto que el código que se genera tras la compilación es un código intermedio o *bytecode*. Este código puede ser, a su vez, interpretado por una máquina virtual instalada en cualquier equipo. Tienen una ejecución lenta, pero su versatilidad para poder ejecutarse en cualquier entorno los hace muy apreciados.

a)

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo");  
    }  
}
```

b)

```
# Hola mundo en Python  
print ("¡Hola mundo!")
```

c)

```
using namespace std;  
int main(int argc, char *argv[]) {  
    std::cout << "Hola mundo" << endl;  
    return 0;  
}
```

d)

```
<script type="text/javascript">  
    alert("Hola Mundo!");  
</script>
```

e)

```
<?php echo '<p>Hola Mundo</p>'; ?>
```

6. Realiza los siguientes pasos:

a) JDK <https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>

b) Apache NetBeans 19  
<https://netbeans.apache.org/download/index.html>

c) Realiza capturas de pantalla y comenta el proceso de obtención del código ejecutable del siguiente programa en Java:

```
public class HolaMundo {  
  
    public static void main(String[] args) {  
        // TODO code application logic here  
        System.out.println("Hola Mundo");  
    }  
}
```