****

**TEORÍA PROGRAMACIÓN**

 ***Actividades de refuerzo***

**1. Se pueden programar satélites, vehículos, electrodomésticos, videoconsolas, cadenas de montaje, robots..., y hasta las atracciones de un parque de recreo. Elige tres de estas cosas y pon ejemplos de programas que deberán contener.**

Satélites: se programa la captura de imágenes y de datos de los distintos sensores, el envío de los mismos a la tierra, el control de la energía y los motores para el movimiento del satélite, etc.

Vehículos: en la actualidad están apareciendo los vehículos sin conductor en los que se programa mediante un ordenador la captura de los datos de los sensores (detección de otros vehículos, detección de las bandas laterales y del carril para circular, detección de obstáculos), se programan la velocidad óptima del vehículo, el control de la dirección, etc.

Electrodomésticos: en los más modernos, como por ejemplo algunas lavadoras, se detectan el tipo de ropa, el color, el grado de suciedad, el peso, etc., y en función de estos parámetros se aplica un programa de limpieza específico.

**2. ¿En qué se diferencian el código fuente y el código objeto?**

Los programas se escriben en lo que se denomina **código fuente**, que son las instrucciones que el programador indica que deben ser ejecutadas a fin de realizar las acciones para las que el programa ha sido diseñado. Sin embargo, este código no lo puede comprender directamente el ordenador y debe ser traducido al único lenguaje que el ordenador puede interpretar: el **lenguaje máquina** o **código binario**. Para realizar esta transformación del código se utilizan unos traductores, llamados **compiladores** e **intérpretes**, que convierten las instrucciones dadas por el programador en instrucciones comprensibles por un ordenador y generan el llamado **código objeto**, que es el archivo traducido a lenguaje máquina.

**3. Explica la diferencia entre un compilador y un intérprete. Explica también la diferencia entre compilar y ejecutar un programa.**

El compilador traduce todo el archivo fuente a código objeto y muestra los posibles errores en un informe posterior.

El intérprete traduce línea a línea el código fuente a código máquina y se detiene si encuentra un error.

Mientras que el intérprete lo lee línea a línea y se detiene en los errores, el compilador lee todo el programa y lo traduce.

El intérprete no genera un archivo ejecutable y el compilador sí, pero es específico para una plataforma.

**4. ¿Qué es la depuración de un programa? ¿Qué tres procedimientos de depuración hemos visto en la unidad?**

La **depuración** de un programa es la quinta fase en el proceso de programación y resulta fundamental a la hora de disminuir costes de implantación, mejorar el rendimiento y garantizar el correcto funcionamiento de los sistemas. Podemos distinguir los siguientes procedimientos de depuración, sabiendo que un programador los usará todos:

1. **Detección de errores de compilación**. Se buscan los errores existentes en un programa y se corrigen para que funcione. Un programador siempre agradece que el lenguaje de programación tenga ayudas a la depuración de código.

2. **Detección de errores en el funcionamiento**. Aunque el programa no tenga errores de sintaxis, puede no estar haciendo lo que el programador desea. Se debe revisar la funcionalidad hasta conseguir que haga las operaciones deseadas.

3. **Optimización de la programación**. Aunque el programa no tenga errores y funcione de manera correcta, siempre hay que buscar la manera óptima de realizarlo, es decir, lograr que la ejecución sea más rápida, que consuma menos memoria, que utilice menos recursos...

**5. ¿Qué tipos de programación existen?**

**• Programación modular.** Consta de varias secciones o módulos que interactúan a través de llamadas a procedimientos. El módulo principal, encargado de llamar al resto de los módulos, integra el programa en su totalidad.

**• Programación estructurada.** Está compuesta por un conjunto de técnicas evolucionadas que aumentan la productividad del programa. Se basa en un proceso lineal y sencillo y se apoya en estructuras secuenciales, selectivas y repetitivas.

**• Programación orientada a objetos.** Usa objetos y sus interacciones para diseñar aplicaciones. Los objetos son entidades que poseen una serie de propiedades destinadas a que los programas sean más fáciles de escribir, mantener y reutilizar.

**6. Enumera y explica los seis pasos que requiere el desarrollo de un programa.**

El desarrollo de un programa requiere los siguientes pasos:

1. Definición y análisis del problema. Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios.

2. Diseño del algoritmo. Dibujar el algoritmo que ha de seguir nuestro programa para contemplar todas las opciones y no olvidarnos ninguna a la hora de programar, así como tener claro cuál será el proceso para escribir código.

3. Codificación del programa (obtenemos el código fuente). Escribir los bloques de instrucciones.

4. Compilación (obtenemos el código objeto). Traducir todo el archivo fuente a código objeto y mostrar los posibles errores en un informe.

5. Depuración de errores y verificación del programa.

6. Explotación (documentación y mantenimiento). Documentar el programa y porque se ha seguido esa procedimiento de programación para que en un futuro sea más fácil realizar modificaciones o correcciones.