**Tarea 4: Repaso Unidad + Lenguajes de programación**

**Desarrollo**

1. Indica la función principal de los siguientes softwares de programación:
   1. Editores de código o texto:  diseñado específicamente para editar el código fuente de programas informáticos.
   2. Compiladores: tipo de traductor que transforma un programa entero de un lenguaje de programación a otro.
   3. Depuradores: es un programa usado para probar y depurar (eliminar) los errores de otros programas
   4. Enlazadores:  programa que toma los objetos generados en los primeros pasos del proceso de compilación, la información de todos los recursos necesarios, quita aquellos recursos que no necesita, y enlaza el código objeto con sus bibliotecas con lo que finalmente produce un fichero ejecutable o una biblioteca.
   5. Interpretadores o traductores:  programa informático capaz de analizar y ejecutar otros programas.
2. Completa la siguiente tabla comparativa entre Java, C++ y Python.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Java | C++ | Python |
| Utilizado para crear | Desarrollar aplicaciones | Nubes, videojuegos, bibliotecas… | Inteligencia artificial, desarrollo web… |
| Pros | Independiente de la plataforma.  Fácil de aprender.  Orientado a objetos. | Alto rendimiento.  Lenguaje actualizado.  Multiplataforma. | Simplificado y rápido.  Elegante y flexible.  Ordenado y limpio.  Portable. |
| Contras | Rendimiento menor.  Solo se puede usar si disponemos de una máquina virtual.  Sintaxis más engorrosa. | Lenguaje muy amplio.  Una compilación por plataforma.  Manejo de librerías más complicado. | La curva de aprendizaje no es tan sencilla.  La mayoría de los servidores no tienen soporte a Python.  Algunas librerías que trae no son del gusto de la comunidad. |

1. Indica las características principales y los lenguajes soportados de los siguientes entornos de programación de escritorio:
   1. Eclipse: Dispone de un editor de texto con un analizador sintáctico. La compilación es en tiempo real.

**Lenguajes soportados:** Java, ANSI C, C++, JSP, sh, perl, php, sed

* 1. NetBeans: Gestión de interfaz y configuración de usuario, gestión de almacenamiento, gestión de ventana, marco asistente, librería visual de Netbeans y herramientas de desarrollo integrado.

**Lenguajes soportados:** Java, JSP, Haskell, CPP, yacc, sh, lex, perl

* 1. Visual Studio: Subrayados ondulados y acciones rápidas, limpieza de código, refactorización, intellisense, cuadro de búsqueda…

**Lenguajes soportados:** C++, C#, java, Python…

* 1. VS Code: Resaltado de sintaxis, snippets, autocompletado de código, refactorización, depuración…

**Lenguajes soportados:** Java, Go, C, C++, Python, PHP, JavaScript…

1. Busca e indica 2 editores de código online e indica qué lenguajes soporta cada uno.

Visual Code Studio: soporta C#, F#, visual basic, PHP, Python, Perl, SQL, Bash y java.

Notepad++: soporta C, C++, java, C#, XML, HTML, PHP, javascript, ASCII, ASP, SQL, CSS, Perl, Python y Lua.

1. Indica los criterios o factores que se deben de tener en cuenta a la hora de seleccionar un lenguaje de programación.

Deberás tener en cuenta varios factores dependiendo de: la portabilidad del lenguaje (cuanto más portable sea, mejor), la optimización del lenguaje, y la velocidad.

1. Indica qué lenguaje(s) seleccionarías para cada opción, e indica el motivo:
   1. Móvil: Java o Kotlin para android y para IOS swift o objective-C.
   2. Aplicaciones de escritorio: visual C++, visual basic o java,
   3. Ciencias: R, Python, SQL, Java, Scala, Julia o MATLAB.
   4. Desarrollo web
      1. Frontend: HTML
      2. Backend: Java, PHP, Python, Javascript.
2. ¿Podríamos programar sin lenguaje de programación? Argumenta tu respuesta.

Si podríamos programar ya que existe el movimiento NoCode que consiste en una serie de plataformas capaces de responder cualquier necesidad que puedas tener a la hora de hacer un desarrollo de software. Las plataformas NoCode utilizan interfaces gráficas en la que puedes ver lo que haces en vez de escribirlo mediante un lenguaje.

**Test**

1. Tipo de lenguajes en el que el diseño de los programas se basa más en los datos y su estructura; es decir, que la unidad de proceso es el objeto y en él se incluyen los datos (variables) y operaciones que actúan sobre ellos:
   1. Declarativos
   2. Imperativos
   3. **Orientados a objetos**
2. Sistema de códigos interpretable directamente por un circuito microprogramable que se compone de un conjunto de instrucciones que determinan acciones que serán realizadas por la máquina:
   1. Lenguaje de bajo nivel
   2. **Lenguaje máquina**
   3. Lenguaje de alto nivel
3. Un intérprete traduce los símbolos de un lenguaje de programación a su equivalente escrito en lenguaje máquina (proceso conocido como compilar); por último, se obtiene un programa ejecutable:
   1. **Verdadero**
   2. Falso
4. Lenguaje de programación que proporciona poca capacidad de servicio o funcionamiento del microprocesador de una computadora:
   1. **Lenguaje de bajo nivel**
   2. Lenguaje máquina
   3. Lenguaje de alto nivel
5. Herramienta que permite desarrollar software o programas cuyo objetivo principal es el de la solución de problemas en una computadora:
   1. **Lenguaje de programación**
   2. Diseño web
   3. Base de datos
6. Lenguaje que se caracteriza porque su estructura semántica es muy similar a la forma como escriben los humanos, lo que permite codificar los algoritmos de manera más natural, en lugar de codificarlos en el lenguaje binario de las máquinas o a nivel de lenguaje ensamblador:
   1. Lenguaje de bajo nivel
   2. **Lenguaje de alto nivel**
   3. Lenguaje máquina
7. Programa que analiza la información, la ejecuta en un código fuente, toma y procesa un código, lo traduce y lo ejecuta:
   1. Compilador
   2. **Intérprete**
   3. Sintaxis
8. Las fases de la compilación son:
   1. Creación de código fuente y compilación del programa
   2. Compilación del programa
   3. Creación de código fuente y enlace del programa con las funciones necesarias de la biblioteca.
   4. **Todas las opciones anteriores son incorrectas.**
9. ¿Cómo se llama el proceso de traducción de código fuente a código objeto?
   1. Interpretación
   2. Ejecución
   3. **Compilación**
10. 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 cascada el más recomendable?
    1. **No**
    2. Sí
11. El compilador recibe código fuente y genera:
    1. **Código máquina**
    2. Código objeto
    3. Ninguna de las anteriores
12. Relaciona los tipos de códigos con su característica más relevante:
    1. Código ejecutable **f**
    2. Código objeto **e**
    3. Código fuente **d**
    4. Escrito en lenguaje máquina pero no ejecutable
    5. Escrito en algún lenguaje de programación de alto nivel, pero no ejecutable
    6. Escrito en lenguaje máquina y directamente ejecutable.
13. ¿En qué modelos el resultado de una fase es la entrada de la siguiente?
    1. Modelo en V
    2. **Modelo en cascada**
    3. Todos los anteriores
14. ¿Cuál es la fase más importante del desarrollo de software?
    1. **Análisis**
    2. Codificación
    3. Diseño
15. Indica la opción falsa sobre el modelo en cascada:
    1. Es un modelo bastante rígido
    2. **Las fases pueden realizarse sin seguir un orden.**
    3. Identifica las fases principales del desarrollo de software.
16. Indica la opción correcta sobre Las metodologías ágiles:
    1. Tienen pocos roles y artefactos
    2. Grupos pequeños
    3. El cliente es parte del equipo
    4. **Todas son verdaderas.**
17. ¿Cuál de las siguientes opciones sobre SCRUM es falsa?
    1. Desarrollo iterativo
    2. **Ciclos largos de trabajo.**
    3. Flexibilidad
18. El rol de Scrum Master no se encarga de…
    1. Facilitar el proceso Scrum
    2. Resolver impedimentos
    3. **Definir objetivos del proyecto**