**Universidad central de Venezuela**

**Alumno: Rubén Aparicio**

**Asignatura: programación (0790)**

**Tarea Nº1**  
  
  
Conceptos a investigar:

**ALU**:

Siglas en ingles de arithmetic Logic Unit o unidad aritmerico-logica. Se trata de un circuito digital que realiza operaciones aritméticas (sumas, restas) operaciones lógicas (si, no, y, o) entre los valores de los argumentos. Este procesador aumenta la velocidad de operación del equipo porque tiene la habilidad de realizar operaciones de cómputo más rápidas de forma eficiente.

Un microprocesador moderno (mainframes) pueden tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución y cada una de ellas con múltiples ALU.

**Diferencias entre lenguaje compilado e interpretado**:

Los lenguajes de programación se peuden dividir en dos macrocategorias dependiendo de como se comunican con el procesador.

En un lenguaje compilado, las funciones como print, son convertidas enensamblador por el compilador y posteriormente ensambladas en lenguaje binario para que la PC pueda leerlas.

Los lenguajes compilados son convertidos directamente a código maquina que el procesador puede ejecutar, esto suele hacer que los programas sean mas rápidos y eficientes. Aunque estos necesitan recompilarse cada vez que hay un cambio. Ejemplos de lenguajes compilados son: C, C++, Rust, Go.

El lenguaje interpretado esta representado por un archivo ya escrito en binario, que, una vez lanzado se lee directamente y se ejecuta.

En los lenguajes interpretados, el programa se ejecuta línea por línea y la vez cada comando. E un principio solían ser mas lentos, pero gracias a la “compilación justo a tiempo” se han optimizado bastante en los últimos años. Ejemplos de lenguajes interpretados son: Javascript, Python, Ruby.

**Compilación justo a tiempo**:

Durante la ejecución del programa puede ser compilado a código nativo para mejorar su ejecución.

**Analogía**:

Una buena analogía para entender la diferencia entre un lenguaje compilado e interpretado puede ser una situación en la que queremos cocinar algo siguiendo una receta que no esta en nuestro idioma nativo. En un lenguaje compilado tendríamos primero traducir dicha receta (compilar) antes de proceder a realizarla (ejecutar el programa). En este caso, al hacer la receta tendríamos que seguir paso a paso lo que hemos traducido; y, de haber un cambio, habría que modificar la receta original volver a traducir todo de nuevo y realizar la receta desde le principio nuevamente.

En el caso de los lenguajes interpretados, podríamos imaginar que tenemos la misma receta en un idioma desconocido para nosotros, pero esta vez contamos con un amigo que conoce dicho idioma y nos puede ir traduciendo paso a paso a medida que realizamos la receta (interpretando). En caso de haber modificaciones se puede detener el proceso, elaborar cambios en la receta y reanudar el proceso nuevamente tomando en cuenta las modificaciones.