Nombre: Jonattan Abraham

Apellidos: Aguilar Nájera

Materia: Laboratorio II

Profesor: José Monterroso

Programación

 La programación informática es el arte del proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer.

La programación informática es una de las habilidades esenciales que aprendes cuando estudias informática.

Detrás de todos los programas informáticos que conocemos y usamos de manera cotidiana para facilitarnos diversas actividades de nuestro día con día, existe todo un proceso para poderlos crear. Este proceso es conocido como programación, conozcamos un poco más sobre lo que conlleva este proceso.

Por medio de la programación se establecen los pasos a seguir para la creación del código fuente de los diversos programas informáticos.

Este código le indicara al programa informático que tiene que hacer y como realizarlo.

La programación se guía por una serie de normas y un conjunto de órdenes, instrucciones y expresiones que tienden a ser semejantes a una lengua natural acotada. Por lo cual recibe el nombre de lenguaje de programación. Y así como en los idiomas también en la informática existen diversos lenguajes de programación.

​

Hablando de forma más técnica, la programación se realiza mediante el uso de algoritmos, que se podrían explicar cómo reglas o instrucciones que deben seguirse para resolver el problema y lograr el objetivo.

Algunas de ellas pueden agruparse y de ese modo recibir un nombre para tener la facilidad de ser invocadas con facilidad tantas veces como sea necesario.

El objetivo final de la programación es la de crear software, el cual después será ejecutado de manera directa por el hardware de la computadora, o a través de otro programa, para desempeñar diversas labores para las cuales fue diseñado.



Su funcion

Para crear un [programa](https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) y que la computadora lo interprete y ejecute, las instrucciones deben escribirse en un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n).

El lenguaje entendido por una computadora se conoce como [código máquina](https://es.wikipedia.org/wiki/C%C3%B3digo_m%C3%A1quina). Consiste en secuencias de instrucciones básicas que el procesador reconoce, codificadas como cadenas de números 1 y 0 ([sistema binario](https://es.wikipedia.org/wiki/Sistema_binario)). En los primeros tiempos de la computación se programaba directamente en código máquina. Escribir programas así resultaba demasiado complicado, también era difícil entenderlos y mantenerlos una vez escritos. Con el tiempo, se fueron desarrollando herramientas para facilitar el trabajo.

Los primeros científicos que trabajaron en el área decidieron reemplazar las secuencias de unos y ceros por [mnemónicos](https://es.wikipedia.org/wiki/Mnem%C3%B3nico), que son abreviaturas en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s) de la función que cumple una instrucción de procesador. Por ejemplo, para sumar se podría usar la letra A de la palabra inglesa *add* (sumar). Crearon así una familia de lenguajes de mayor nivel, que se conocen como [lenguaje ensamblador](https://es.wikipedia.org/wiki/Lenguaje_ensamblador) o simplemente ensamblador (en inglés, *assembly*). Con el tiempo los ensambladores incorporaron facilidades adicionales, pero siempre manteniendo una correspondencia directa con las instrucciones de procesador. A nivel conceptual, entonces, programar en ensamblador es muy similar a hacerlo en lenguaje máquina, solo que de una forma más amigable.

A medida que la complejidad de las tareas que realizaban las computadoras aumentaba, el lenguaje ensamblador fue mostrando limitaciones. Para hacer un programa había que conocer en detalle el funcionamiento de la computadora donde se iba a ejecutar, qué instrucciones proveía y cómo emplearlas. A veces las instrucciones eran demasiado básicas, por ejemplo podía haber una para sumar dos números pero no para multiplicar, y entonces era necesario programar un algoritmo que realizara la multiplicación en base a instrucciones más básicas. Otras veces, la forma de emplear las instrucciones era engorrosa. Además, si se usaba otro modelo de computadora, en muchos casos había que reescribir el programa con otras instrucciones. El siguiente paso fue crear los [lenguajes de alto nivel](https://es.wikipedia.org/wiki/Lenguaje_de_alto_nivel).

Una vez que se termina de escribir un programa, es necesario de alguna forma traducirlo a lenguaje máquina, que es lo único que entiende el procesador. Esta tarea es automática, por medio de un programa adicional que toma el código escrito y lo procesa. Hay distintos enfoques para este procesamiento. El enfoque clásico se llama [compilación](https://es.wikipedia.org/wiki/Compilador): el programa toma el código en un lenguaje y genera código en el otro; al programa traductor se lo llama compilador. En general se habla de compilación y compiladores cuando el lenguaje de origen es de alto nivel; si la traducción es desde lenguaje ensamblador, se llama ensamblado y el programa se llama ensamblador (hay que distinguir el lenguaje ensamblador del programa ensamblador; en inglés es más claro, son *assembly language* y *assembler* respectivamente).[2](https://es.wikipedia.org/wiki/Programaci%C3%B3n#cite_note-2)​Generalmente existe una fase posterior a la compilación denominada [enlace](https://es.wikipedia.org/wiki/Enlazador) o enlazado (*linking* en inglés). Los programas pueden escribirse en partes separadas y además pueden usar recursos provistos por [bibliotecas](https://es.wikipedia.org/wiki/Biblioteca_%28inform%C3%A1tica%29). El enlazado, realizado por un programa llamado enlazador, combina todos los componentes y así genera un programa ejecutable completo.

En algunos lenguajes de programación, puede usarse un enfoque diferente que no requiera compilación y enlace: un programa llamado [intérprete](https://es.wikipedia.org/wiki/Int%C3%A9rprete_%28inform%C3%A1tica%29) va leyendo el código y realizando en el momento las acciones que haría el programa. Se evita generar código separado y la experiencia es que se está ejecutando