MODULO I: CONCEPTOS INTRODUCTORIOS

En este módulo desarrollaremos algunos de los conceptos fundamentales que debes conocer para comenzar tu camino en el mundo de la programación informática. El objetivo principal de la presente guía, es el de presentarte sintéticamente, desde conceptos técnicos hasta apreciaciones personales que te servirán a la hora de comenzar a formarte como programador/a. Comencemos..

Computadora: Hardware y Software. La herramienta indispensable del programador.

Antes de comenzar a transitar nuestro camino, será necesario entender cuál es nuestra principal herramienta de trabajo y el sector en el que nos desenvolveremos.

Si analizamos nuestro entorno, podremos identificar varios artefactos que son denominados “inteligentes”, algunos de ellos son: los smartphones, los aires acondicionados, los smartwatchs, etc. Se les atribuye tal nombre, al estar computarizados. Sin embargo, cada uno de ellos, operan de manera diferente y cada uno cumple distintas funciones en la cotidaneidad.

Para entender la diferencia de cada uno de estos artefactos y por consiguiente, poder abordar algunas areas de la informatica, es necesario preguntarnos *¿qué es una computadora?*

Una computadora es un objeto que surge de la sumatoria de dos partes bien diferenciadas, que se necesitan entre si para poder operar, estas partes son denominadas como *hardware y software.* Si bien es cierto que se puede programar desde un dispositivo móvil, por motivos de comodidad y de rendimiento, el programador usualmente utiliza lo que se conoce comercialmente como computadora de escritorio o portátil para desempeñar su trabajo. Partiendo de este concepto, podemos enfocarnos en esas partes que la conforman:

- ➔ Hardware: Se identifica así, a todo el grupo de componentes materiales y físicos que conforman una computadora. Abarca toda la parte tangible de ella(todo lo que se puede tocar), por ejemplo, el teclado, la memoria ram, el gabinete, etc. Este grupo, a su vez, se subdivide en dos subgrupos: el hardware interno, y el hardware periférico (de entrada y/o de

salida). En el mercado laboral y academico, existen perfiles que se especializan en el mantenimiento de esta area como por ejemplo los “tecnicos en reparacion de PC”.

➔ Software: Es el conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación. Es la parte lógica, intangible (que NO se puede tocar), de la computadora. Se subdivide en tres grupos:

- Software de sistema (ej.: sistemas operativos).
- Software de aplicación (ej.: utilitarios, como el procesador de texto Word).
- Software de programación (IDE’s).

El oficio del programador informatico o desarrollador de software, se enmarca en el area del *Software.* A su vez dentro de esta, encontrarás más areas en las que podras especializarte y orientar un perfil, como por ejemplo el area de *Desarrollo de Aplicaciones Movil, Desarrollo de Aplicaciones de Escritorio, Desarrollo Web*, etc.

¿Cuál es la tarea del programador?

La programación informática aborda todo el proceso de diseño, codificación, mantenimiento y depuración de un programa o aplicación informática, por medio de un código fuente.

Nuestra tarea principal como programadores sera definir instrucciones para que un ordenador pueda ejecutar todos esos pasos que tendran como finalidad, automatizar una tarea para el usuario(por ejemplo: una agenda virtual, o una calculadora). Para lograrlo, se definiran algoritmos, en un determinado lenguaje de programación.

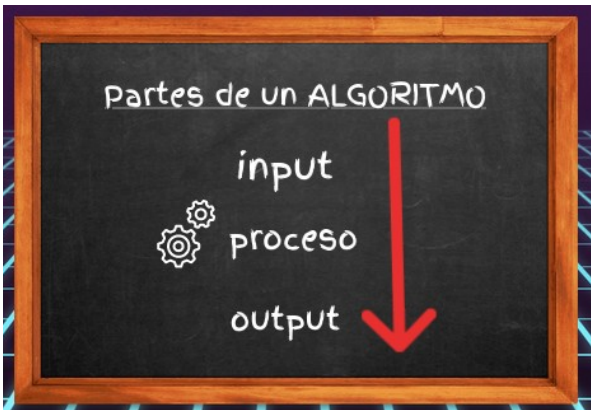
¿Qué es un programa? ¿Y un algoritmo?

Un programa es un conjunto de uno o mas algoritmos, que trabajan de forma ordenada, a fin de cumplir con determinado propósito, realizar una tarea o trabajos específicos, en una computadora.

Un algoritmo, es un conjunto ordenado de instrucciones finitas bien definidas que resuelven un problema. Un manual de instrucciones para colgar una televisión en la pared, una receta de cocina para preparar un postre, las indicaciones de un GPS para llegar a destino, son algunos de las tareas diarias que conforman algoritmos, ya que cumplen con ciertas características que hacen a su esencia. Todo algoritmo debe ser:

- Preciso: Sin ambigüedad.
- Ordenado: Constituir una secuencia clara y precisa, un paso a paso, para poder llegar a un determinado resultado.
- Finito: Ser un número determinado de pasos.
- Concreto: Ofrecer una solución especifica a un problema especifico.
- Definido: El mismo algoritmo debe dar el mismo resultado al recibir la misma entrada.

Componentes o partes de un algoritmo en informatica:



➔ Input(entrada): recoleccion de datos que aportaremos al algoritmo.

➔ Proceso: conjunto de pasos para que, a partir de los datos de entrada, llegue a la solución de la situación.

➔ Output(salida): resultado, a partir de la transformación de los valores de entrada durante el proceso.

Cuando se te presente un problema al que debas darle solución, podrás representar tu algoritmo de distintas maneras, para asegurarte que la solución que proporcionas es la ideal:

- Diagrama de flujo: Es una representación gráfica de un algoritmo, normalmente se utiliza el lenguaje UML (Lenguaje unificado de modelado) , puedes utilizar plantuml para llevarla adelante.
- Pseudocódigo: Este busca acercarse aun mas a un algoritmo en un lenguaje de alto nivel, pero, en un lenguaje mucho más amigable para el programador.
- Código fuente: Lo podemos definir como una serie de instrucciones secuenciales, escritas en un lenguaje de programación determinado que, a través de un compilador o intérprete, puede ser ejecutado en una máquina.

La importancia de la lógica de programación.

Como vimos anteriormente, dar una solucion a un problema de manera eficaz, implica ser muy conscientes a la hora de definir la secuencia de instrucciones y qué instrucciones le daremos a la computadora. **De ello depende el resultado que obtendremos.**

Pensemos que estamos configurando un robot, y queremos programarlo para que pueda lavarse los dientes. Los pasos que el autómata deberia llevar adelante son:

1. Tomar la pasta de dientes con la mano derecha.
2. Con la mano izquierda destapar la pasta de dientes.
3. Dejar la tapa en el lavamanos
4. Con la mano izquierda ya liberada, tomar el cepillo de dientes
5. Posicionar la pasta de dientes sobre las cerdas del cepillo
6. Depositar la pasta en el cepillo: Apretar la pasta de dientes mientras la deslizamos sobre las cerdas, y soltar cuando corresponda para que no caiga al piso
7. Introducir el cepillo en la boca
8. Cepillar: Mover el cepillo de izquierda a derecha cuando este posicionado en cada diente

etc…..

Este es un algoritmo que apunta a resolver esa tarea. Ahora bien, ¿qué pasaría si movemos una instrucción de lugar? Imaginemos que lo establecemos de la siguiente manera:

1.Introducir el cepillo en la boca

2.Tomar la pasta de dientes con la mano derecha.

3.Con la mano izquierda destapar la pasta de dientes.

4.Dejar la tapa en el lavamanos

5.Con la mano izquierda ya liberada, tomar el cepillo de dientes

6.Posicionar la pasta de dientes sobre las cerdas del cepillo

7.Depositar la pasta en el cepillo: Apretar la pasta de dientes mientras la deslizamos sobre las cerdas, y soltar cuando corresponda para que no caiga al piso

8.Cepillar: Mover el cepillo de izquierda a derecha cuando este posicionado en cada diente

En esta situación, seguramente al intentar ejecutar la primer instrucción, el autómata te dirá que no puede hacerlo, ya que no podra ingresar un objeto a su boca sin antes haberlo tomado. Lo mismo sucedería si obviamos algun paso fundamental para la resolucion de la tarea, por ejemplo, si

nunca le damos la instrucción de destapar la pasta el automata, al presionarla haria estallar el recipiente.

Si damos instrucciones claras, bien organizadas, estaremos estableciendo una “logica adecuada” que apunte a solucionar el problema; el arte de pensar y definir bien ese paso a paso, se desarrolla con la practica.

Para conseguir un primer acercamiento a la logica de programación, puedes jugar con LightBot , alli comprenderás que tanto influye la logica de nuestras soluciones.

Un poco de historia..

Ada Lovelace, es considerada la pionera de la programación informática, ya que fue ella quien creó el primer algoritmo informático. Gracias a su invención, a lo largo de la historia hemos podido desarrollar tres tipos de lenguajes de programación. Ellos son:

- Lenguaje de máquina: El lenguaje que los ordenadores reconocen, por lo que aún hoy todo lenguaje es convertido a este. Basado en el sistema binario(0 y 1).
- Lenguaje ensamblador: Se utilizaban palabras simples, mnemónicas y abreviaturas que tenían su correlativo y eran traducidas al código máquina.
- Lenguaje de alto nivel: con el Fortran se dio inicio a la aparición de lenguajes basados en conjuntos de algoritmos mucho más complejos. Es un idioma artificial prediseñado formado por signos, palabras y símbolos que permite la comunicación entre el programador y el ordenador. Algunas ventajas: Más cercano a un lenguaje humano. Más fácil de programar. Menos posibilidad de cometer errores. Permiten la portabilidad.

En este grupo de estudio, nos enfocaremos en aprender la sintaxis básica del lenguaje de programación de alto nivel Python 3.10.

Comentario final: Las habilidades del programador.

Si partimos de una visión holística de la profesión, lograremos entender los cómo y por qué del oficio del programador.

Cuando una persona se capacita en distintas herramientas informáticas(como lo puede ser un lenguaje de programación de alto nivel, un tipo de base de datos, etc.) , se dice que está formando o desarrollando sus *habilidades duras o “hard skills”*. Las hard skills, refieren a todo el *conocimiento y habilidad técnica* que se le demanda al programador para desempeñarse en determinado puesto.

Retomando el concepto de que el programador viene a resolver problemas, podemos deducir que al profesional también se le demandarán ciertas *habilidades interpersonales*,que determinarán en mayor o menor medida, su forma de desenvolverse en el ámbito informático. Estas son, las llamadas *habilidades blandas o "soft skills"*.

Éstas ultimas, son tan importantes como las primeras. Requieren de mucha introspección y autocrítica. Notarás que algunas de ellas las tendrás mas o menos desarrolladas que otras, pero la buena noticia, es que una vez que las identifiques, podrás trabajar en ellas. Podemos mencionar algunas, por ejemplo: la constancia, la independencia, la tolerancia a la frustración, la capacidad de trabajar en equipo, la buena comunicación, etc.

Algunos consejos...

Si bien apuntaremos a desarrollar habilidades duras(nos instruiremos en Python), te dejo algunos consejos que considero necesarios destacar antes de que inicies tu camino en el mundo de la programación:

1. Aprender a programar depende de cuánto practiques, sin **constancia** no asimilarás los conceptos teóricos. Como bien dice el dicho "la practica, hace al maestro".
2. La programación no es para genios, es para resilientes. Rara vez, las cosas funcionan a la primera, como programador deberás ser firme cuando tengas un problema, y volver a intentarlo hasta llegar a la solución. No importa cuanto tiempo te lleve resolverlo.
3. La **curiosidad** será tu mejor aliada. Las herramientas son muchas, y están en constante evolución. Como programador, deberás mantenerte en constante aprendizaje.
4. El conocimiento surge en la **cooperación**. A través del **trabajo en equipo**, reforzaras saberes y adquirirás otros.
5. Saber investigar es fundamental. Todas aquellas dudas, incluso la resolucion de errores, la encontraras en la web. Todo programador debe ser en mayor o menor medida **autodidacta,** es decir, con el tiempo deberás desarrollar cierta independencia que te permita entender cada vez que se le presente un problema.

En el próximo modulo, comenzaremos a instruirnos en una herramienta fundamental para todo programador, que no solo te permitirá organizar y almacenar tus proyectos o prácticas, si no que te ayudará a compartirlo con otras personas y hasta colaborar con ellas; sin dudas, te servirá para demostrar tus avances en programación.