

Fundamentos de Programación

Por qué es importante aprender a programar

Vivimos en un mundo cada vez más digitalizado y tecnológico y atravesado por la programación. Ésta es muy importante y forma parte de nuestras vidas, incide en ella en forma transversal.

Prácticamente todos los sectores de la sociedad utilizan software y esto hace que la demanda de profesionales de sistemas e informática crezca día a día.

Steve Jobs dijo: *“Todo el mundo en este país debería aprender a programar porque te enseña a pensar”*.

En esta década la demanda de programadores está creciendo exponencialmente, actualmente en el país y la región no hay suficiente RRHH para abastecer la demanda del sector. Existen por un lado empresas locales que exportan software para todo el mundo y profesionales que trabajan para empresas extranjeras.

Nos sobran los motivos

La importancia de la programación radica en diversos motivos: agiliza procesos, incrementa el alcance de las ventas de distintos productos, amplía la propuesta educativa de universidades hacia regiones que eran impensadas (este fenómeno se ha acentuado a partir de la pandemia), permite operar a través de homebanking, permite hacer home office, otorga turnos a través de sistemas en bancos, análisis clínicos, entre otros.

Ejemplos

- Compramos en tiendas en línea.
- Conocemos personas a través de redes sociales.
- Nos comunicamos a través de WhatsApp, Telegram, Signal, Slack, Skype, Google Meet, etc.
- Hacemos listas en Trello.
- Escuchamos música en Spotify.
- Jugamos videojuegos (no solo aplicado a gamificación sino también se utiliza actualmente en salud, ambiente, educación).
- Alquilamos hoteles/casas a través de Airbnb, Booking, Despegar, Mercado Libre, etc.
- Miramos películas y series en Netflix, Amazon, Flow, etc.
- Accedemos a información de nuestra salud en nuestro celular a través de distintas apps (Cuidar, miArgentina, etc.).

¿Qué ejemplos pueden brindar ustedes?

Comenzamos...

La computadora es una máquina que por sí sola no puede hacer nada, necesita ser programada, es decir, debe recibir instrucciones u órdenes que le digan lo que tiene que hacer.

El proceso de programación consiste en dado un problema, el programador debe idear una solución y expresarla usando un algoritmo, luego debe codificarlo en un determinado lenguaje de programación y por último ejecutar el programa en la computadora, como resultado obtenemos una solución al problema inicial.

Conceptos Básicos

Algoritmo

Un algoritmo es una secuencia de pasos, ordenados que alcanzan un fin determinado, una tarea o un resultado. Un mismo problema puede ser resuelto a través de distintos algoritmos.

Programa

Es un conjunto de instrucciones que se escribe en un lenguaje de programación para que la computadora pueda realizar una actividad.

Programación

Disciplina que permite la resolución de problemas mediante la presentación de una serie de pasos que codificados en un lenguaje de programación pueden ser ejecutados en una computadora.

Lenguaje de Programación

Conjunto de reglas, símbolos y palabras especiales utilizadas para construir un programa.

Lenguaje Máquina

Lenguaje utilizado por la computadora, compuesto por instrucciones codificadas en sistema binario. (0 y 1).

Compilador

Programa que traduce un programa escrito en lenguaje de alto nivel a lenguaje de máquina.

Etapas simplificadas en el desarrollo de una solución por computadora

Antes de iniciar el desarrollo de un programa, es prioritario entender el problema a resolver mediante un detallado análisis, luego representarlo mediante Pseudocódigo o diagrama de flujo y finalmente escribirlo en un lenguaje de programación adecuado.



Planteo de un problema

Para resolver un problema con el uso de la computadora, debemos entender bien el mismo en su contexto, analizando las posibles entradas y salidas que va a requerir el programa.

Problema

Calcular el sueldo mensual de un trabajador conociendo la cantidad de horas trabajadas y el pago por hora.

Solución:

El problema requiere de una fórmula específica, *sueldo = horas trabajadas * costo hora*. Se ingresan valores como entrada y se obtienen valores enteros o reales de salida.

Planteo

ENTRADA

Valor de entrada 1 al programa = cantidad de horas trabajadas = CHT

Tipo de dato Numérico Entero o Real

Valor de entrada 2 al programa = pago por hora = PPH

Tipo de dato Numérico Real

PROCESO

Sueldo = CHT * PPH

SALIDA

Sueldo

Pasos o Etapas para la resolución de un problema por medio de la computadora

1. Entender el problema y analizarlo.

Esta fase está dada por el enunciado del problema, debe ser definido en forma clara y precisa. Es importante que se conozca lo que se desea que realice la computadora. Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

- Los datos de entrada. (ENTRADA)
- Los métodos y fórmulas que se necesitan para procesar los datos. (PROCESO)
- La información que se desea producir (SALIDA).

2. Diseñar una estrategia.

3. Diseñar un algoritmo (Pseudocódigo, Diagrama de flujo).

4. Refinar el algoritmo (Prueba de escritorio)

5. Codificar el algoritmo

6. Ejecutar el programa

Características de un buen algoritmo

- Ser Preciso. Debe definirse de manera rigurosa, sin dar lugar a ambigüedades.
- Ser Definido. Si se sigue un algoritmo dos veces, se obtendrá el mismo resultado.
- Ser Finito. Debe terminar en algún momento.
- Puede tener cero o más elementos de entrada.
- Producir al menos un resultado. Los datos de salida serán los resultados de efectuar las instrucciones.

A tener en cuenta: Entre dos algoritmos que lleven a un mismo objetivo, siempre será preferible el que insuma menos recursos (se deberá analizar la optimización de tiempos e instrucciones).

Prueba de escritorio o Depuración

Es la comprobación o prueba que se hace del funcionamiento de un algoritmo para saber si está bien hecho. Consiste en tomar datos específicos como entrada y seguir la secuencia indicada en el algoritmo hasta obtener un resultado. El análisis de estos resultados indicará si el algoritmo es correcto o si por el contrario hay necesidad de corregirlo o hacerle ajustes.

Programación Estructurada

Es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora recurriendo únicamente a tres estructuras de control básicas: secuencia, selección e iteración.

Constante

Una constante es un dato numérico o alfanumérico que no cambia durante la ejecución del programa. Es necesario inicializar las constantes al principio del algoritmo. Esto determina el tipo de la constante.

Ejemplos

- PI=3.14
- SALUDO="Hola mundo"
- MAXIMO=500
- RESPUESTA='S'

Variable

Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa. Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

Ejemplo: $\text{area} = \text{base} * \text{altura} / 2$

Las variables son: área, base y altura

- Una variable es un objeto de datos que posee un valor conocido por su nombre (identificador)
- Toda variable tiene un tipo asociado

-
- Su valor puede ser modificado, pero en un instante dado sólo contiene un valor determinado
 - Es necesario declarar las variables al principio del algoritmo
 - Elegir el nombre mediante un identificador
 - Determinar el tipo de datos que va a contener

Tipos De Datos

Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un carácter, tal como 'b' o '35', un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.

Simple

- Numéricos
- Lógicos
- Carácter

Estructurados

- Cadena (String)
- Arreglos
- Registros
- Archivos

Identificadores

Los identificadores representan los datos de un programa (constantes, variables, tipos de datos). Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria de la computadora, que permite acceder a su contenido.

Reglas para formar un identificador

- Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no deben contener espacios en blanco.
- Letras, dígitos y caracteres como guion bajo están permitidos después del primer carácter.
- El tamaño de los identificadores puede ser de varios caracteres.
- El nombre del identificador debe ser significativo.

Operadores

Operadores

Lógicos

- OR ó lógico
- AND y lógico
- NOT no

Aritméticos

- + Suma
- - Resta
- * Multiplicación
- / División real
- DIV División entera
- MOD Resto de la división entera

Relacionales

- = Igual
- <> Distinto
- < Menor
- > Mayor
- <= ó =< Menor o igual
- >= ó => Mayor o igual

Operando (Operador) Operando
 ┌───────────┴───────────┐
 Valor
 (constante o variable)

Clasificación de los problemas elementales

- Problemas de evaluación: los resultados son normalmente alfanuméricos o numéricos
- Problemas de decisión: los resultados son verdadero o falso

Estrategia

Plan de acción, bosquejo, se divide el problema en subproblemas para darle solución.

Representación de un Algoritmo

- Diagrama de flujo
- Pseudocódigo

Para realizar un programa es conveniente el diseño o definición previa del algoritmo.

El diseño de algoritmos requiere creatividad y conocimientos profundos de la técnica de programación.

Luis Joyanes Aguilar, programador experto y autor de muchos libros sobre lógica y programación nos dice **“en la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de**

programación es sólo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo”.

Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación. El algoritmo es la infraestructura de cualquier solución, escrita luego en cualquier lenguaje de programación.

Algoritmo para cambiar un foco

- Para retirar el foco quemado
 1. Colocar una escalera debajo del foco
 2. Subir la escalera
 3. Desenroscar el foco en el sentido contrario de las agujas del reloj
 4. Bajar la escalera
- Poner un foco nuevo
 1. Tomar el foco nuevo
 2. Subir la escalera
 3. Enroscar el foco en el sentido de las agujas del reloj
 4. Bajar la escalera
 5. Retirar la escalera

Pseudocódigo

Es la técnica que permite expresar la solución de un problema mediante un algoritmo escrito en palabras normales de un idioma (por ejemplo, el castellano), utilizando palabras imperativas. El Pseudocódigo utiliza palabras como: leer, mostrar, calcular.

Estructura de un algoritmo

ALGORITMO (identificador)

AMBITO: declaración de variables, tipos y constantes

CUERPO: lista de instrucciones/ acciones

FIN

Acciones

- Simples (leer, asignación, mostrar)
- Estructuradas - Condicional (simple, alternativo, anidado, múltiple) - Cíclicas, repetitivas o iterativas) (para, mientras, repetir...hasta)
- Procedimientos y funciones

Expresiones

Una expresión es la combinación de distintas variables, constantes, valores constantes, operadores, paréntesis y nombres de funciones. Toda expresión tiene en todo momento un valor concreto que es el resultado de evaluarla en el sentido de izquierda a derecha, siempre teniendo en cuenta la precedencia de los operadores y los paréntesis que aparecen en la expresión. El valor devuelto tras la evaluación de una expresión determina el tipo de la expresión (entera, lógica, etc.).

Ejemplos

- $5+6-5*2$
- $(5+6-5) * 2$
- $(A<0) \text{ OR } B \text{ AND } C$