**¿QUE ES UN PROGRAMA?**

A lo largo del tiempo se ha entendido al tema de programación como el hecho de conocer un lenguaje y a travez de él lograr la solución a determinados problemas.

Este concepto hoy se ha modificado, entendiendose como programación a la posibilidad de la resolución del problema, independientemente del lenguaje a utilizar.

La solución de problemas por medio de la programación es una tarea que requiere meditación, planificación, lógica y perseverancia.por lo cuál es una actividad que tiene características de mucha creatividad y desafio personal.

PROGRAMA 🡪 Conjunto de instrucciones

PARADIGMA 🡪 Modelo o forma de ver la realidad. Van cambiando a lo largo del tiempo o conviven

PARADIGMA DE PROGRAMACION 🡪 Genera modelos

MODELO 🡪 Representación en escala de menor calidad o costo

Para la resolución de un problema, es necesario primero entenderlo, luego crear el **algorítmo** necesario y por último volcarlo a un lenguaje determinado. Es aquí donde se ve que no basta conocer sólo un lenguaje sino que es necesario conocer una técnica de programación.

**Algoritmo**, es un conjunto finito de pasos, procesados en un tiempo finito despues del cuál se obtiene la solución deseada.

O sea que todo algoritmo debe tener un principio y un fin determinado.

Existen etapas para la solución del problema, las cuales podriamos definir como la estrategia a seguir

* Determinar datos
* Determinar resultados
* Determinar los procesos que debemos aplicarle a esos datos para obtener esos resultados

Los **datos** serán todos aquellos elementos que provengan del exterior de nuestro programa, y que su procesamiento en una forma determinada nos permitirá obtener los resultados requeridos que es lo que llamamos **información.**

# REPRESENTACIÓN DE LOS DATOS Y DE LOS RESULTADOS EN UN PROGRAMA

Los datos son representados por medio de variables. Estas llevaran nombres claros y representativos de lo que contienen y podrán ser de distintos tipos. Por ejemplo:

|  |  |  |
| --- | --- | --- |
| **Dato o resultado** | **Variable** | **Tipo** |
| Nombre del empleado | nomemp | Texto |
| Horas trabajadas | hstrab | Numérica |
| Valor hora | valhs | Numérica |
| Sueldo | sdo | Numérica |
| Fecha de nacimiento | fnac | Fecha y hora |
| Comentarios | curric | Memo |

Los tipos de datos estarán ligados al lenguaje a utilizar pero en general todos tienen los mismos tipos o muy parecidos.

Dentro de las variables se pueden guardar datos, valores constantes y a otras variables que vayan surgiendo durante la ejecución del algoritmo.

A continuación los datos se verán afectados por procedimientos para obtener un resultado, es conveniente dividir estos procedimientos en procesos lo más pequeños posible a fin de tener mayor control sobre los mismos, y que una posterior modificación en una parte de él no repercuta en todo el resto negativamente, “*divide y triunfaras*“.

Lo que restaría es armar el algoritmo que no es más que colocar estos pequeños procesos en una forma lógicamente ordenada para poder obtener los resultados deseados.

# TEOREMA DE BOHM Y JACOPINI

El teorema de Böhm y Jacopini establece, que un **programa propio** puede ser escrito utilizando únicamente tres tipos de estructuras de control, las cuales son:

* Estructuras secuenciales
* Estructuras de selección
* Estructuras de repetición

Para que la programación sea estructurada, los programas han de ser propios. Un programa se define como **propio** si cumple las siguientes condiciones:

* Si tiene un solo punto de entrada y un solo punto de salida de cada estructura
* Si todas las sentencias del algoritmo son alcanzables, esto es, existe al menos un camino que va desde el inicio hasta el fin del algoritmo
* Si no posee ciclos infinitos

De este **teorema** se deduce que, si los algoritmos se diseñan empleando exclusivamente dichas estructuras de control, los algoritmos, y por consecuencia los programas derivados de ellos, serán propios.

# TIPOS DE ESTRUCTURAS

Dentro de la programación estructurada reconocemos tres estructuras básicas

* Estructuras secuenciales
* Estructuras condicionales
* Estructuras iterativas o de repetición

# 

# DIAGRAMACION RESPETANDO EL PARADIGMA ESTRUCTURADO

Esta es una forma más de representar la lógica de programación.

Un proceso es una secuencia de instrucciones que ocupan una cantidad de recursos del computador y permiten la solución de un problema, ya sea en función de un solo proceso o de varios subprocesos que al combinarse, correcta y lógicamente, generan la solución deseada.

La diagramación que utilizaremos consiste en dibujar a todos sus elementos, que se encuentran ordenados en forma secuencial de arriba hacia abajo.

En todos los casos dentro del símbolo se escribe la instrucción, el procedimiento que se llama, etc. por ejemplo

emp, sdo

Proceso1

Sdo > 15000

Sdo = ch \* vh

Simbología definida para el desarrollo del diagrama de flujo bajo el estándar ANSI.

|  |  |
| --- | --- |
| **SÍMBOLO** | **SIGNIFICADO** |
|  | **Inicio** y **Fin** del Diagrama |
|  | **Entrada de Datos**. (Se homogeniza el uso de este símbolo para entradas desde cualquier origen incluyendo el teclado. En caso de ser necesario identificar el origen se aclara dentro del símbolo) |
|  | **Proceso**. Indica una acción o proceso. (p.e . cambio de valor de variables, asignaciones y operaciones matemáticas) |
|  | **Llamada a Subrutina** |
|  | **Salida**. (Se homogeniza el uso de este símbolo para salidas hacia cualquier destino Pantalla o Impresora. En caso de ser necesario identificar el destino se aclara dentro del símbolo.) |
|  | **Decisión**. Compara dos datos y dependiendo de su resultado lógico (verdadero o falso) se toma la decisión de seguir por un camino del diagrama u otro. |
|  | **Conector (in page)**. Enlaza dos partes del programa de la misma página. Si el diagrama tuviera varios, se distinguen por el número que llevan en su interior. |
|  | **Conector (out page)**. Indica el enlace de dos partes de un programa en distintas páginas. |
| **EL SIGUIENTE SÍMBOLO SOLE SE APLICA A LOS DIAGRAMAS DENOMINADOS DE ALTA ABSTRACCIÓN PARA SIMPLIFICAR LA COMPRENSIÓN DEL USO DE ESTRUCTURAS ITERTIVAS** | |
|  | **Iteración**. Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces. |

**CUESTIONARIO**

1. Realice un cuadro sinóptico donde se indique las distintas etapas de la evolución de los lenguajes de programación, el paradigma al cual responde y los lenguajes característicos de ese paradigma
2. ¿Qué enuncia el teorema de Böhm y Jacopini?
3. Resuma que entiende por diagrama de flujo, organigrama, ordinograma y seudocódigo
4. ¿Cómo se escribe un comentario de una línea en C?
5. ¿Cómo se escribe un comentario de varias líneas en C?
6. Realice un cuadro indicando como se define cada tipo de dato en C
7. ¿Para qué sirve una variable auxiliar?
8. ¿Qué es un contador? Normalmente, ¿de qué tipo de datos es?
9. ¿Qué es un acumulador?
10. ¿Qué diferencia existe entre un contador y un acumulador?