Informática I
INF 140

Capítulo 1 - *Algoritmos*

1.1 - *Introducción*

- Un **algoritmo** es un conjunto finito de instrucciones que especifican la secuencia de operaciones a realizar, en orden, para resolver un problema determinado.
- En otras palabras es una fórmula para resolver un problema.
- Generalmente es una lista de la siguiente forma:

  Paso 1: “Hacer algo”
  Paso 2: “Hacer algo”
  Paso 3: “Hacer algo”
  ⋮
  Paso n: “Hacer algo”

Capítulo 1 - *Algoritmos*

1.1 - *Introducción*

- Un alumno solicita ser el ayudante de un ramo. El profesor examina en la base de datos de la escuela el historial del alumno. Si el alumno está capacitado el profesor lo acepta como ayudante, en caso contrario la solicitud del alumno será rechazada.

- Los pasos del algoritmo en descripción narrativa son los siguientes:

  1. Inicio
  2. Leer solicitud del alumno
  3. Leer historial del alumno
  4. Si, el alumno está capacitado, el profesor acepta la solicitud, en caso contrario la solicitud es rechazada.
  5. Fin

Capítulo 1 - *Algoritmos*

1.1 - *Introducción*

- Calcular el promedio de 3 notas.

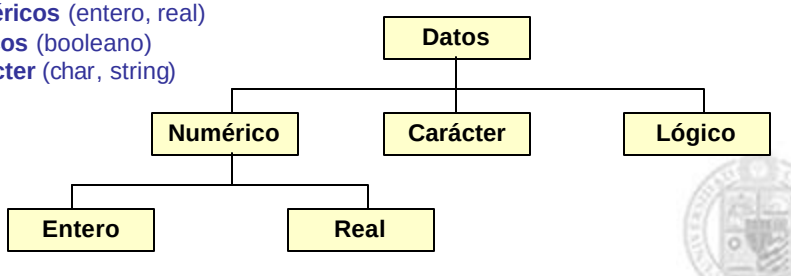
  1. Inicio
  2. Leer nota 1
  3. Leer nota 2
  3. Leer nota 3
  4. Asignar a suma_de_notas el resultado de nota1+ nota2 + nota3.
  5. Asignar a promedio el resultado de suma_de_notas / 3
  6. Fin

```
real: nota1,nota2,nota3,promedio
inicio
  leer (nota1)
  leer (nota2)
  leer (nota3)
  suma_de_notas <- nota1 + nota2 + nota3
  promedio <- suma_de_notas / 3
fin
```

Capítulo 1 - *Algoritmos*

1.2 - *Tipos de Datos*

- El principal objetivo de todo computador es el manejo de información o datos. Un **dato** es la expresión general que describe los objetos con los cuales opera una computadora.
- Existen dos clases de tipos de datos: simples(sin estructura) y compuestos(estructurados)
- Los tipo de datos simples son los siguientes:
- **Numéricos** (entero, real)
  Lógicos (booleano)
  Carácter (char, string)



Capítulo 1 - *Algoritmos*

1.3 - *Variables, constantes e identificadores*

- Una **constante** es un objeto o una partida de datos cuyo valor no cambia durante el desarrollo del algoritmo o ejecución del programa.
- Una **variable** es un objeto o una partida de datos cuyo valor puede cambiar durante el desarrollo del algoritmo o ejecución del programa.
- Un **identificador** es el nombre que posee la variable. No se deben utilizar como identificadores palabras reservadas del lenguaje de programación.

1.4 - *Expresiones*

- Las **expresiones** son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

+ , - , * , / , ^ , div, mod

Capítulo 1 - *Algoritmos*

1.5 - *Estructuras de control*

- Las **estructuras de control** son las que permiten conducir el flujo del programa, existen dos tipos de estructuras de control las estructuras selectivas y las estructuras repetitivas.

1.5.1 – *Estructuras selectivas*

- Las **estructuras selectivas** se utilizan para tomar decisiones lógicas, también son llamadas **estructuras de decisión o alternativas**.

1.5.1.1 – *Alternativa simple*

```
si <condicion>
  <accion 1>
  <accion 2>

  <accion n>
fin_si
```

```
si (numero MOD 2 = 0)
  escribir("número es par")
fin_si
```

Capítulo 1 - *Algoritmos*

1.5.1.2 – *Alternativa doble*

```
si <condicion>
  <accion 1>

  <accion n>
si_no
  <accion 1>

  <accion n>
fin_si
```

```
si (numero MOD 2 = 0)
  escribir("número es par")
si_no
  escribir("número es impar")
fin_si
```

1.5.1.3 – *Alternativa multiple*

```
segun_sea expresion(E) hacer
  e1: <accion 1>
  e2: <accion 2>
  e3: <accion 3>

  en: <accion n>
fin_si
```

```
segun_sea dia MOD 7 hacer
  1: escribir("el día es Lunes")
  2: escribir("el día es Martes")
  3: escribir("el día es Miércoles")
  4: escribir("el día es Jueves")
  5: escribir("el día es Viernes")
  6: escribir("el día es Sábado")
  0: escribir("el día es Domingo")
fin_si
```

Capítulo 1 - *Algoritmos*

1.5.2 – *Estructuras repetitivas*

- Las **estructuras repetitivas** se utilizan para repetir una o varias acciones un número determinado de veces.

1.5.2.1 – *Mientras*

```
mientras <condicion>
  <accion 1>
  <accion 2>

  <accion n>
fin_mientras
```

```
mientras (i< 10)
  escribir("el número es:", i)
  i=i+1
fin_mientras
```

1.5.2.2 – *Repetir*

```
repetir
  <accion 1>
  <accion 2>

  <accion n>
hasta <condicion>
```

```
repetir
  escribir("el número es:", i)
  i=i+1
fin_repetir (i = 10)
```

Capítulo 1 - *Algoritmos*

1.5.2.3 – *Desde*

```
desde v<-vi hasta vf
  <accion 1>
  <accion 2>

  <accion n>
fin_desde
```

```
Desde i=1 hasta 10
  escribir("el número es:", i)
  i=i+1
fin_desde
```