PROGRAMACIÓN

Algoritmos: Repaso

2022

Lic. Mariela A. Velázquez

¿Qué es un algoritmo?

Un Algoritmo es una descripción de la forma en que se debe realizar una tarea o un proceso, en una secuencia finita de pasos que se llevarán a cabo en un tiempo finito.

El diseño del algoritmo se realiza usando un PSEUDOCÓDIGO.

- Neutro: Es independiente al lenguaje que se vaya a usar.
- Completo: Permite expresar cualquier idea computacional.

Características del pseudocódigo

Cada Algoritmo

- Tiene un nombre que indica que tarea resolverá.
- Puede tener o no una entrada: datos que necesitamos que el usuario ingrese para realizar un acción u operación.
- Tiene una salida: datos que devuelve nuestro programa, en este caso nuestro algoritmo.
- Toda entrada se debe LEER: todos los datos "ingresados" por nuestros usuarios se deben leer.
- Se puede asignar valores a las variables.
- Toda salida se debe ESCRIBIR: todos los datos que obtenemos de nuestro algoritmo se deben "mostrar".
- Para indicar el final de un algoritmo, hacemos uso de la acción primitiva **PARAR()**

ALGORITMO: pagoLuz
ENTRADAS: consumo: reales, vencida: entero (0: no / 1:si), cliente: entero positivo,
metodo_pago: entero (0: tarjeta/1: contado),
SALIDA: importe: reales.
V. AUX: excede: reales
CONSTANTES:

A1. LEER(cliente)
A2. **calcular_importe**
A3. PARAR

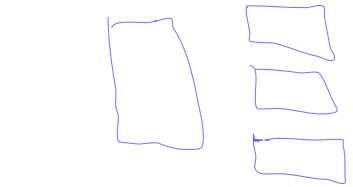
cant ← 0
escribir(cant)

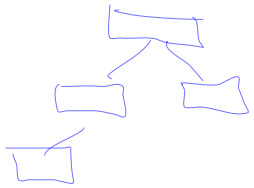
Técnicas de Diseño

- El proceso de adicionar más detalles a una solución de un problema se conoce como **REFINAMIENTO SUCESIVO.**

- El método o técnica: **DIVIDE AND CONQUER** con la que se ataca un problema tiene la característica de ser una técnica **TOP-DOWN.**

- Es una estrategia que permite descomponer jerárquicamente un problema largo y complejo en subproblemas más pequeños y fáciles de resolver que el problema de partida.





Ejemplo

Enunciado:

La modalidad de pago de la factura de luz en una cierta ciudad es la siguiente: se establece una tarifa mensual para el consumo mínimo (hasta los 100 kwh) de $115. Si se ha sobrepasado dicho consumo, se suma una tarifa de $72 por cada kwh adicional; pero si está vencida la factura, la tarifa que se usa ya no es de $72 sino de $100. Además en cualquier caso se hace un descuento del 5% al monto total, por pago de contado. Diseñe un algoritmo con niveles de refinamiento, que determine cuánto debe pagar cada cliente. Pruebe la misma para un número indefinido de clientes.

Ejemplo

ALGORITMO: pagoLuz

ENTRADAS: consumo: reales, vencida: entero (0: no / 1:si), cliente: entero positivo,
metodo_pago: entero (0: tarjeta/1: contado),

SALIDA: importe: reales.

V. AUX: excede: reales

CONSTANTES:

A1. LEER(cliente)

A2. **calcular_importe**

A3. PARAR

Ejemplo

```
A3. calcular_importe
MIENTRAS (cliente > 0)
    LEER(consumo, metodo_pago, vencida)
        SI (consumo <= 100) ENTONCES
            importe ← 115
        SINO
            SI(vencida = 1) ENTONCES
                excede ← consumo - 100
                importe ← 115 + excede * 100
            SINO
                excede ← consumo - 100
                importe ← 115 + excede * 72
            FIN SI
        FIN SI
        SI (metodo_pago = 1) ENTONCES
            importe ← importe – (importe * 0.05)
        FIN SI
        ESCRIBIR (importe)
        LEER (cliente)
FIN MIENTRAS
```

Enunciado:

La modalidad de pago de la factura de luz en una cierta ciudad es la siguiente: se establece una tarifa mensual para el consumo mínimo (hasta los 100 kwh) de $115. Si se ha sobrepasado dicho consumo, se suma una tarifa de $72 por cada kwh adicional; pero si está vencida la factura, la tarifa que se usa ya no es de $72 sino de $100. Además en cualquier caso se hace un descuento del 5% al monto total, por pago de contado. Diseñe un algoritmo con niveles de refinamiento, que determine cuánto debe pagar cada cliente. Pruebe la misma para un número indefinido de clientes.

Procedimientos

- Un procedimiento es un subprograma que ejecuta una tarea determinada. El mismo puede o no recibir argumentos o parámetros de cualquier tipo. A diferencia de una función, el procedimiento produce un efecto, no retornar ningún valor
- El algoritmo invoca al procedimiento con el nombre del mismo en una expresión seguida de una lista de argumentos que deben coincidir en cantidad, tipo y orden con los del procedimiento definido.

Sintaxis de procedimientos

PROCEDIMIENTO nombreProc(param1, param2, … $param_i$): tipo1, tipo2,…, $tipo_i$
    Acciones
Fin_procedimiento

```
PROCEDIMIENTO CORREGIR(n1, n2, n3): real, real, real

SI (n1 + n2 + n3  < 5) ENTONCES
        ESCRIBIR("Desaprobado/a")
SINO
        SI ((n1 >= 3,5 * 0,4) ∧ (n2 >= 2,5 * 0,2) ∧ (n3 >= 4 * 0,75)) ENTONCES
                ESCRIBIR("Aprobado/a")
        SINO
                ESCRIBIR("Desaprobado/a")
        FIN SI
FIN SI

FIN PROCEDIMIENTO
```

Ejemplo

La cátedra de Elementos necesita automatizar la corrección de parciales, para esto se requiere un **procedimiento** que a partir de los puntajes individuales (obtenidos por un estudiante) para cada punto de un parcial, se informe si el estudiante aprueba o desaprueba.

Cada punto de un parcial tiene un puntaje de manera que sumados llegan a 10. Se aprueba con 5, pero además es requisito cumplir con un porcentaje mínimo de nota para cada punto, tal como se detalla a continuación:

Punto	Puntaje sobre 10	Porcentaje mínimo para dar por válido
1	3,50	40%
2	2,50	20%
3	4	75%

Ejemplo

```
PROCEDIMIENTO CORREGIR(n1, n2, n3): real, real, real

SI (n1 + n2 + n3  < 5) ENTONCES
        ESCRIBIR("Desaprobado/a")
SINO
        SI ((n1 >= 3,5 * 0,4) ∧ (n2 >= 2,5 * 0,2) ∧ (n3 >= 4 * 0,75)) ENTONCES
                ESCRIBIR("Aprobado/a")
        SINO
                ESCRIBIR("Desaprobado/a")
        FIN SI
FIN SI

FIN PROCEDIMIENTO
```

Funciones

Una función es un subprograma que puede o no recibir argumentos o parámetros, datos de tipo numérico o no numérico, y devuelve un único resultado.

El algoritmo o programa invoca la función con el nombre de esta última en una expresión seguida de una lista de argumentos que deben coincidir en cantidad, tipo y orden con los de la función que fue definida.

Funcion (A, e, Num1)

devuelve 1 resultado

Sintaxis de funciones

FUNCION **nombrefun**(**param1, param2, … param$_i$**): **tipo1, tipo2,…, tipo$_i$** → **tipo**
    Acciones
    RETORNA resultado
Fin_funcion

```c
#include <stdio.h>

int main()
{
    printf("Hasta la próxima clase!!\n");

    return 0;
}
```