

ELEMENTOS DE COMPUTACIÓN Y LÓGICA



Universidad Nacional de Tucumán



2020 - AÑO DEL GENERAL MANUEL BELGRANO

Contenido

Unidad 1. Propositiones Simples y Compuestas. Representación simbólica. Conectivos lógicos: Negación, Conjunción, Disyunción, Condicional, Bicondicional. Tablas de verdad. Jerarquía de conectivos. Clasificación de fórmulas lógicas por su significado: Tautologías, contradicciones y contingencias. Implicación y equivalencia lógica. Leyes Lógicas

Unidad 2. Razonamientos. Componentes: premisas, conclusión. Reglas de inferencia. Métodos de demostración: directo e indirecto. Validez de un razonamiento. Consistencia de Premisas.

Unidad 3. Lógica de Predicados. Funciones proposicionales. Universo del discurso. Representación simbólica. Cuantificadores: Cuantificador universal y cuantificador existencial. Alcance de un cuantificador. Variables libres y variables vinculadas. Propositiones categóricas. Negación de proposiciones cuantificadas. Equivalencia de proposiciones cuantificadas universalmente y existencialmente. Razonamientos. Reglas de especificación universal y existencial. Reglas de generalización universal y existencial.

Unidad 4. Álgebras Booleanas y circuitos combinatorios. Propiedades de los circuitos combinatorios. Funciones Booleanas.

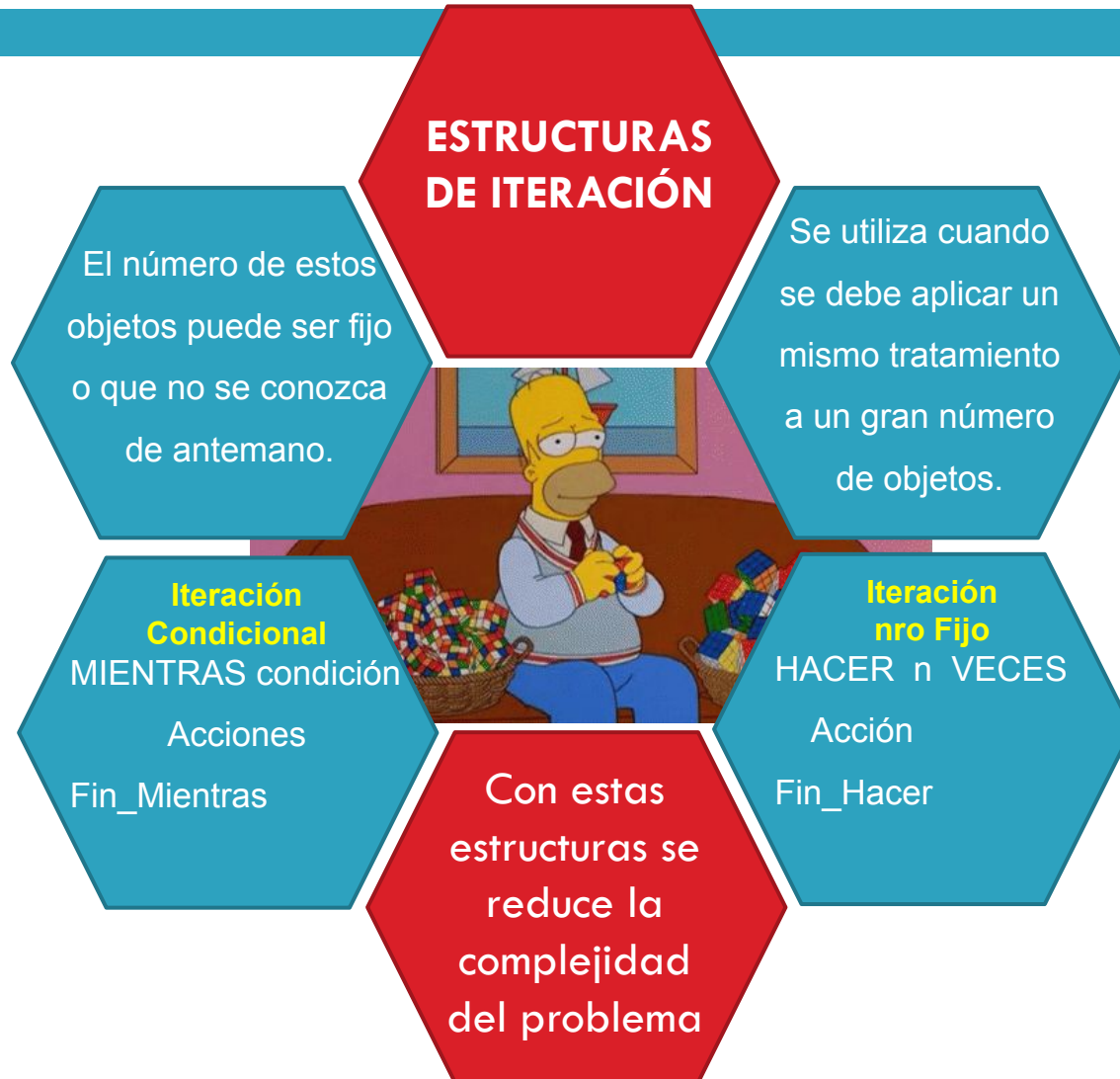
Unidad 5. Sistemas de numeración binario. Conversiones. Suma y resta de binarios. Sistema de numeración octal. conversiones. Sistema de numeración hexadecimal.

Unidad 6. Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, análisis iterativo. Generalización del concepto de procedimiento: Acciones parametrizadas, funciones.

Unidad N° 6 – Parte 3:

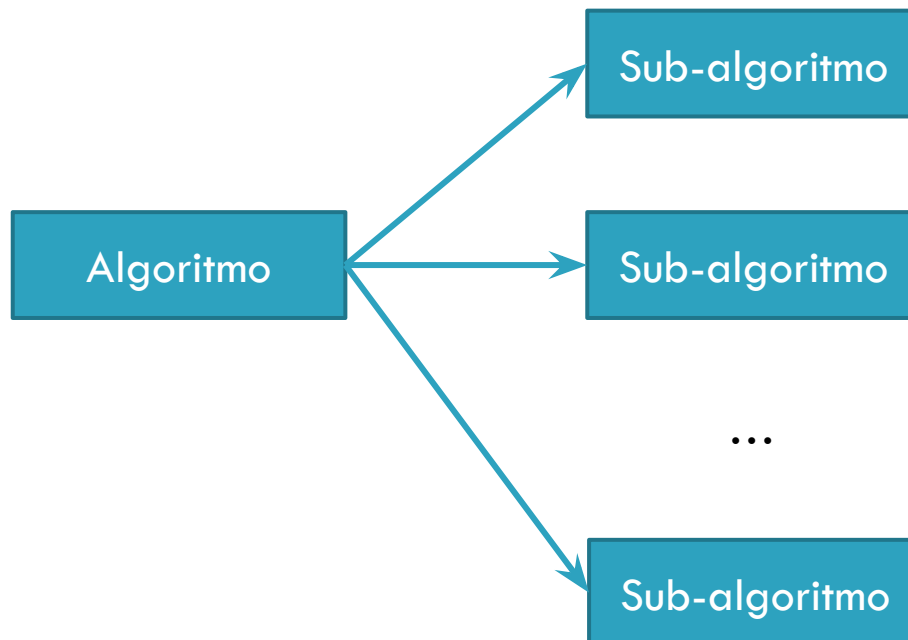
Generalización del concepto de procedimiento. Acciones Parametrizadas, funciones.

REPASEMOS LO VISTO



PROCEDIMIENTOS Y FUNCIONES

La resolución de problemas complejos se facilita considerablemente si se dividen en problemas mas pequeños; y la resolución de estos subproblemas se realiza mediante “sub-algoritmos”.

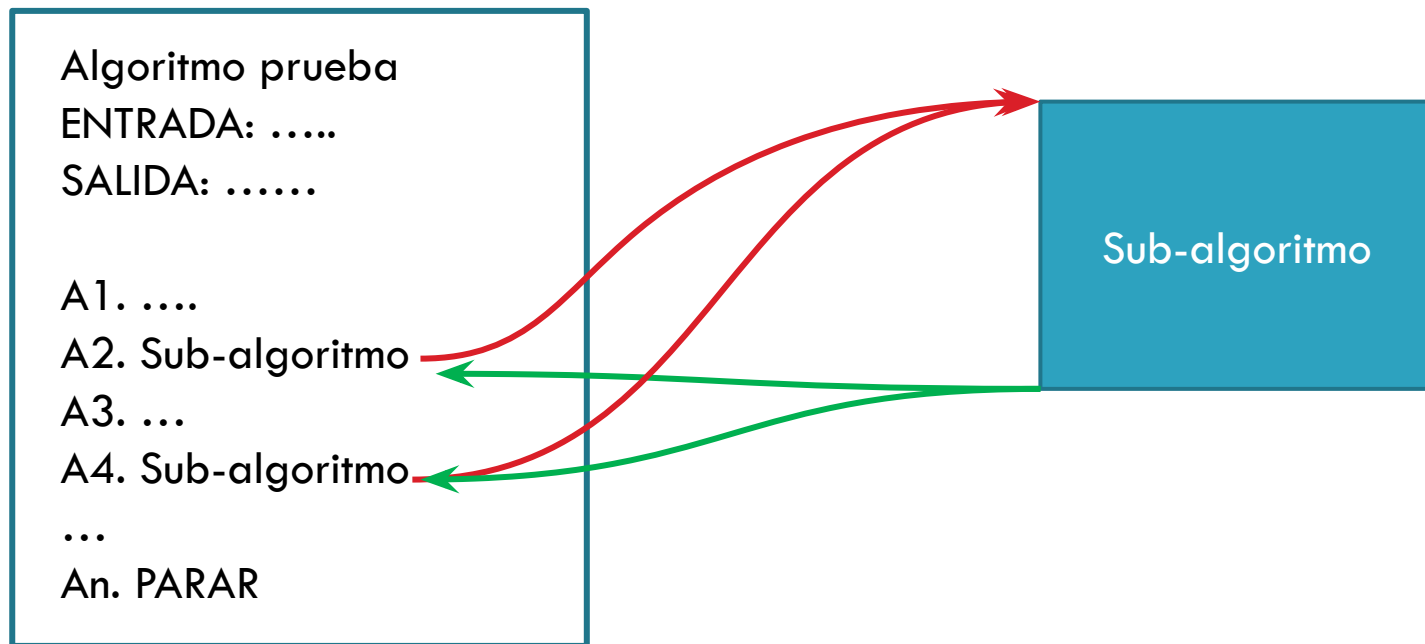


PROCEDIMIENTOS Y FUNCIONES

- Los sub-algoritmos son módulos que están diseñados para ejecutar alguna tarea específica. Éstos pueden ser, funciones o procedimientos.
- Se escriben solamente una vez, pero pueden ser referenciados en diferentes puntos del algoritmo principal, de este modo se puede evitar la duplicación innecesaria del código.
- El modulo principal se ejecuta en una primera instancia, que da la orden de inicio de ejecución de los subprogramas.

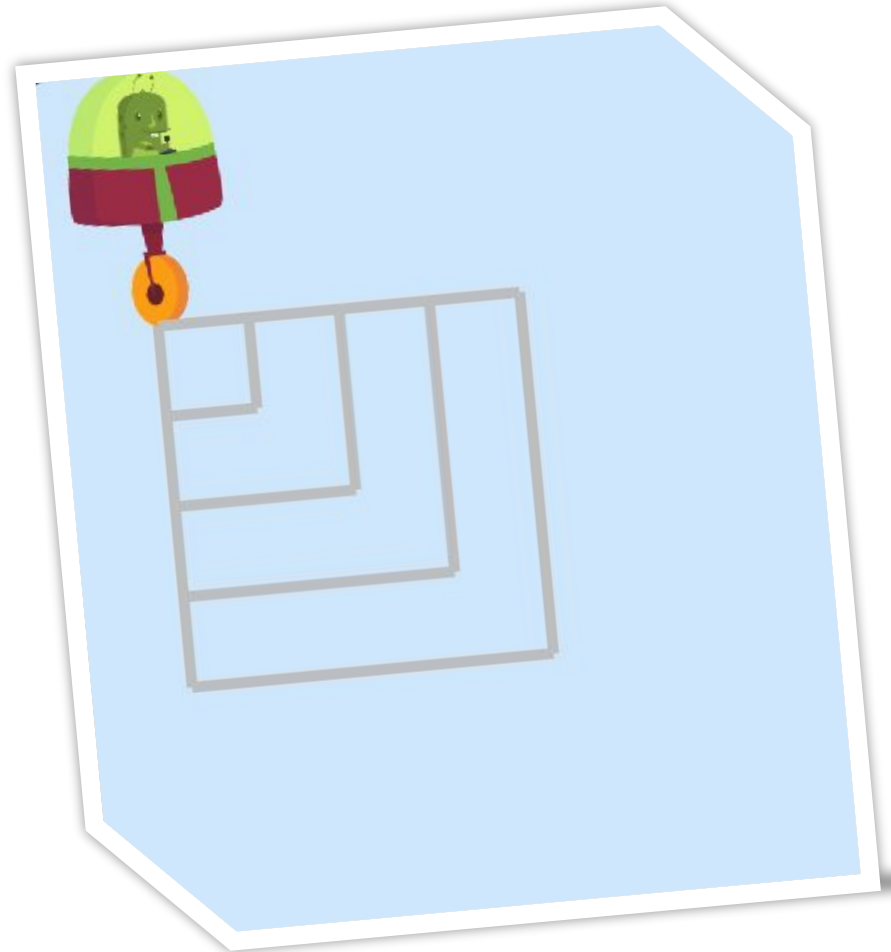
PROCEDIMIENTOS Y FUNCIONES

- El subprograma es un programa en sí mismo, ejecutado por la solicitud del programa principal o de otro subprograma, una n cantidad de veces. Cuando realiza la solicitud, el programa se detiene hasta que el subprograma deja de realizar su tarea, luego continúa; esto se conoce como control de ejecución.



PROCEDIMIENTOS Y FUNCIONES

<https://pilasbloques.program.ar/online/#/desafio/32>



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.

Sintaxis de funciones

```
FUNCION NombreFuncion (parametro1, parametro2, ... parametroN): tipo1, tipo2, ..., tipoN → tipo  
    Acciones  
    RETORNA resultado  
Fin_funcion
```



Al invocarse una función debe ser asignada a una variable del mismo tipo del retorno de la misma. Por ejemplo: `resultado ← NombreFuncion(parametro1, parametro2, ... parametroN)`

FUNCIONES

Sintaxis de funciones

FUNCION **nombrefun**(**param1**, **param2**, ... **param_N**): **tipo1**, **tipo2**, ..., **tipo_N** → **tipo**

Acciones

RETORNA resultado

Fin_funcion

Descripción

- **nombrefun**: es el nombre de la función.
- **param1**, **param2**, ... **param_N**: datos que la función requiere para realizar un proceso determinado.
- **tipo1**, **tipo2**, ..., **tipo_N**: tipos de datos de cada uno de los parámetros.
- **tipo**: es el tipo de datos de la variable que devolverá la función.

FUNCIONES

Ejemplo: Contar la cantidad de vocales presentes en una cadena de caracteres finalizada en punto.

Función Esvocal(letra): carácter \rightarrow booleano

Resultado \leftarrow F

SEGUN letra

"a": Resultado \leftarrow V

"e": Resultado \leftarrow V

"i" : Resultado \leftarrow V

"o": Resultado \leftarrow V

"u": Resultado \leftarrow V

Fin_según

RETORNA Resultado

Fin_función

Algoritmo: cuentavocales

ENTRADA: cc: secuencia de caracteres con MF= '.'

SALIDA: cont : entero positivo

Vble Aux.: vocal: booleano

A1. contar_vocales

A2. ESCRIBIR(cont)

A3. PARAR

A2. contar_vocales

A2.1 cont \leftarrow 0

A2.2 LEER(cc)

A2.3 MIENTRAS (cc \neq MF)

vocal \leftarrow Esvocal(cc)

SI (vocal = V) ENTONCES

cont \leftarrow cont + 1

Fin_si

LEER(cc)

Fin_Mientras

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 y no retorna 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(parametro1, parametro2, ... parametroN): tipo1, tipo2, ..., tipoN  
    Acciones  
Fin_procedimiento
```

PROCEDIMIENTOS

Sintaxis de procedimientos

PROCEDIMIENTO **nombreProc**(**param1**, **param2**, ... **param_N**): **tipo1**, **tipo2**, ..., **tipo_N**

Acciones

Fin_procedimiento

Descripción

- **nombreProc**: es el nombre del procedimiento.
- **param1**, **param2**, ... **param_N**: datos que el procedimiento requiere para **realizar** un proceso determinado.
- **tipo1**, **tipo2**, ..., **tipo_N**: tipos de datos de cada uno de los parámetros.

PROCEDIMIENTOS

Ejemplo: Dado 3 notas de exámenes de un curso de 30 alumnos, se desea saber el promedio de notas de cada uno de los alumnos y la condición en la que finalizan el año. Los estados posibles son: promocionado ($\text{prom} \geq 7$) aprobado ($4 \leq \text{prom} < 7$) o desaprobado ($\text{prom} < 4$)

Algoritmo: Calificaciones

ENTRADA: nota1, nota2, nota3: reales

SALIDA: promedio: real, condicion: cad caract.

A1. HACER 30 VECES

A1.1 LEER(nota1, nota2, nota3)

A1.2 calcularProm(nota1, nota2, nota3)

Fin_Hacer

A4. Parar

PROCEDIMIENTO calcularProm(nota1, nota2, nota3): real, real, real

prom \leftarrow (nota1 + nota2 + nota3) / 3

ESCRIBIR(prom)

Si (prom ≥ 7) ENTONCES

ESCRIBIR("Promocionado/a")

SINO

Si (prom ≥ 4) ENTONCES

ESCRIBIR("Aprobado/a")

SINO

ESCRIBIR("Desaprobado/a")

Fin_Si

Fin_procedimiento

$\neg q$
 $p \wedge p \leftrightarrow q$
 $p \wedge \neg q$
 $p \vee q$
 $p \leftrightarrow q$
 $\neg p$
 $p \square q$
 $\neg q$

Nos vemos en
PROGRAMACIÓN

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F