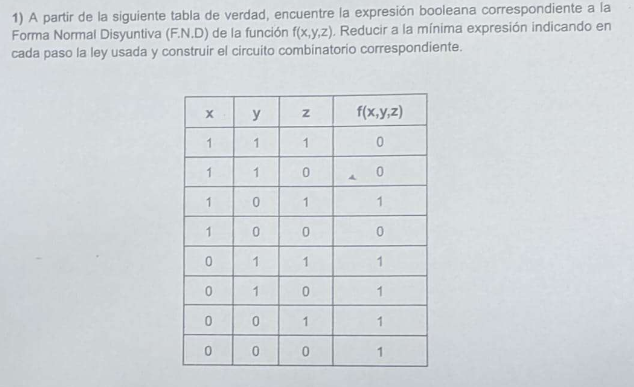
**Práctica y ejercicios para recuperar.**

**\*PARCIAL 2\***



F.N.D:

F: \**Agrupo por factor común.*

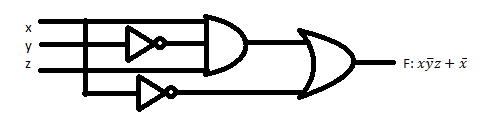
F: *\*Ley de complementación.*

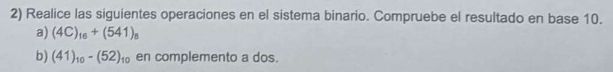
F: *\*Ley de identidad.*

F: *\*Agrupo por factor común.*

F: *\*Ley de complementación.*

F: *\*Ley de identidad.*



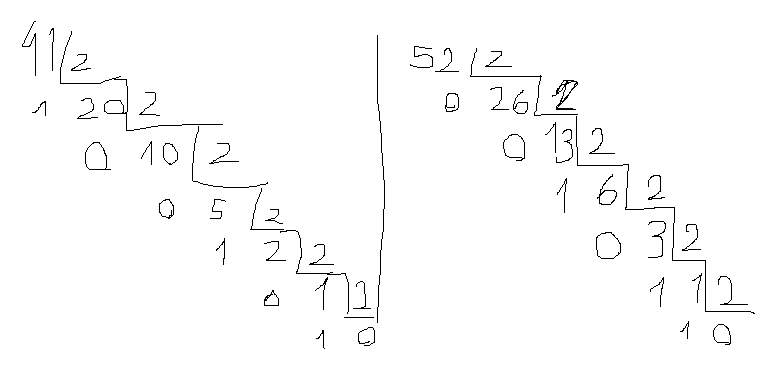


(4C)16 + (541)8 = 0 1 0 0 1 1 0 0 | (76)10

+ 1 0 1 1 0 0 0 0 1 | (353)10

1 1 0 1 0 1 1 0 1 | (429)10

(41)10 – (52)10= 101001 – 110100

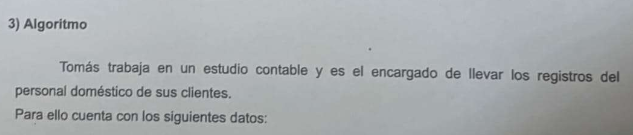


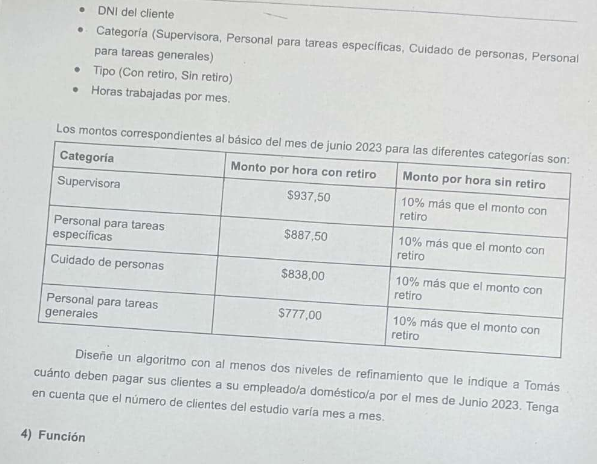
Uso complemento a2 tal que:

0|1 0 1 0 0 1

+1|0 0 1 1 0 0

1|1 1 0 1 0 1 | (complemento a2): 001011 (11)10 que es -11 con a2 por el 1 antes del “|”





**ALGORITMO** Registros

ENTRADA: dni: enteros positivos, categoría: cadena de caracteres{“s”, ”pe”, ”c”, ”pg”}, tipo: cc{“cr”, “sr”}, horas\_trabajadas: reales

SALIDA: precio\_final: reales

CONSTANTES: AUMENTO:1.1; MS: 937.5, MPE: 887.5, MC: 838, MPG: 777

A1. LEER(dni, categoría, tipo, horas\_trabajadas)

A2. Calcular\_horas

A3. Calcular\_tipo

A4. ESCRIBIR(precio\_final)

A5. PARAR

A2. Calcular\_horas

A2.1 SEGÚN categoría:

A2.2 “s”: precio\_final <- MS\*horas\_trabajadas;

A2.3 “pe”: precio\_final <- MPE\*horas\_trabajadas;

A2.4 “c”: precio\_final <- MC\*horas\_trabajadas;

A2.5 “pg”: precio\_final <- MPG\*horas\_trabajadas;

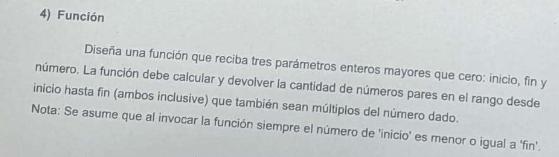
A2.6 FIN SEGÚN

A3. Calcular\_tipo

A3.1 SI (tipo=”sr”) ENTONCES

A3.2 precio\_final <- precio\_final\*AUMENTO;

A3.3 FIN SI



FUNCION pares(inicio, fin, numero): enteros mayores que cero -> enteros mayores que cero

HACER fin VECES (i: inicio…fin)

SI (RESTO(i, numero) = 0) ENTONCES

SI (RESTO(i,2) = 0) ENTONCES

par <- par+1;

FIN\_SI

FIN\_SI

i <- i+1;

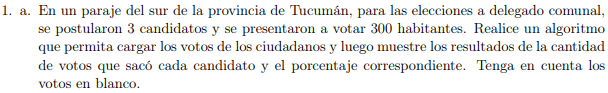
FIN\_HACER

RETORNA par

FIN\_FUNCION

**\*FIN PARCIAL 2\***

**\*Algoritmos\***



ALGORITMO votos

ENTRADA: votos: enteros, candidato: entero{1,2,3}

SALIDA: vc1, vc2, vc3, vb: enteros, pc1, pc2, pc3, pb: reales

VAR. AUX: HABITANTES=300

A1. LEER(votos,candidato)

A2. Calcular\_votos

A3. Calcular\_porcentaje

A4. ESCRIBIR(vc1, vc2, vc3, vb, pc1, pc2, pc3, pb)

A5. PARAR

A2 Calcular\_votos

A2.1 HACER 300 VECES

A2.2 SEGÚN candidato:

A2.3 1: vc1 <- vc1+1;

A2.3 2: vc2 <- vc2+1;

A2.4 3: vc3 <- vc3+1;

A2.5 SINO : vb <- vb+1;

A2.6 FIN\_SEGÚN

A3 Calcular\_porcentaje

A3.1 pc1 <- (vc1/HABITANTES)\*100;

A3.2 pc2 <- (vc2/HABITANTES)\*100;

A3.3 pc3 <- (vc3/HABITANTES)\*100;

A3.4 pb <- (vb/HABITANTES)\*100;



ALGORITMO votos

ENTRADA: votos: enteros, candidato: entero{1: c1,2: c2,3: c2}, habitantes: entero

SALIDA: vc1, vc2, vc3, vb: enteros, pc1, pc2, pc3, pb: reales, ganador: cc

VAR. AUX: i: entero

A1. LEER(votos,candidato, habitantes)

A2. Calcular\_votos

A3. Calcular\_porcentaje

A4. Ver\_ganador

A5. ESCRIBIR(vc1, vc2, vc3, vb, pc1, pc2, pc3, pb, ganador)

A6. PARAR

A2 Calcular\_votos

A2.1 HACER habitantes VECES (i:1. . habitantes)

A2.2 SEGÚN candidato:

A2.3 1: vc1 <- vc1+1;

A2.3 2: vc2 <- vc2+1;

A2.4 3: vc3 <- vc3+1;

A2.5 SINO : vb <- vb+1;

A2.6 FIN\_SEGÚN

A3 Calcular\_porcentaje

A3.1 pc1 <- (vc1/habitantes)\*100;

A3.2 pc2 <- (vc2/habitantes)\*100;

A3.3 pc3 <- (vc3/habitantes)\*100;

A3.4 pb <- (vb/habitantes)\*100;

A4 Ver\_ganador

A4.1 SI (vc1 > vc2 Y vc1 >vc3) ENTONCES

A4.2 ganador <- “c1”

A4.3 SINO SI (vc2 > vc1 Y vc2>vc3) ENTONCES

A4.4 ganador <- “c2”

A4.5 SINO SI (vc3>vc1 Y vc3>vc2) ENTONCES

A4.6 ganador <- “c3”

A4.7 SINO SI (vc1=vc2 Y vc1=vc3) ENTONCES

A4.8 ganador <- “Empate”

A4.9 FIN\_SI



ALGORITMO oración

ENTRADA: oración: cc con MF=”.”

SALIDA: a, e, i, o, u, b, p,v: entero positivo

A1. Reconteo

A2. ESCRIBIR(a,e,i,o,u,b,p,v)

A3. PARAR

A1 Reconteo

A1.1 LEER(oración)

A1.2 MIENTRAS (oración <> MF)

SEGÚN (oración)

“ ”: b <- b+1

“.”: p <- p+1

“a”: a <- a+1

“e”: e <- e+1

“i”: i <- i+1

“o”: o <- o+1

“u”: u <- u+1

FIN\_SEGÚN

A1.3 FIN\_MIENTRAS

A1.4 v <- a+e+i+o+u



ALGORITMO números

ENTRADA: numero: entero

SALIDA: sumatoria: enteros

VAR. AUX: i: entero

A1. Sumatoria\_multiplos

A2. PARAR

A1. Sumatoria\_multiplos

A1.1 HACER 10 VECES

i <- 0

sumatoria <- 0

LEER(numero)

HACER numero VECES (i:1. .numero)

i <- i+1

SI (RESTO(i, 5) = 0) ENTONCES

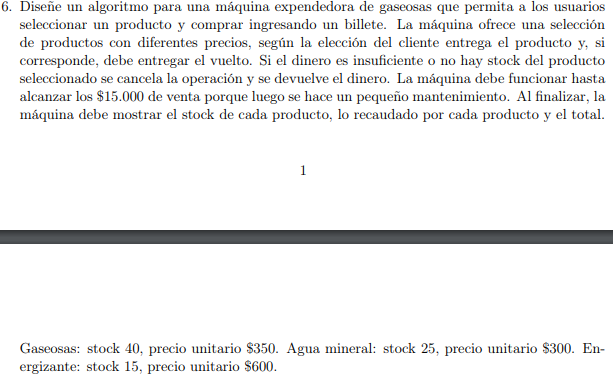
sumatoria <- sumatoria+i

FIN\_SI

ESCRIBIR(sumatoria)

FIN\_HACER

A1.2 FIN\_HACER



ALGORITMO maquina

ENTRADA: billete: entero positivo, elección: entero{0: gaseosa, 1: agua, 2: energizante}

SALIDA: vuelto: entero positivo, producto: cc

CONSTANTES: MAX\_VENTA = 15000, PG = 350, SG = 40, PA=300, SA=25, PE = 600, SE=15

Var. Aux: i: real

A1. Calcular\_venta

A2. PARAR

A1. Calcular\_venta

A1.1 MIENTRAS (i<MAX\_VENTA)

LEER(billete, elección)

SEGÚN elección

0: SI (billete >= PG Y SG>0) ENTONCES

vuelto <- billete-PG

i <- billete-vuelto

SG <- SG-1

ESCRIBIR (vuelto, i, SG)

FIN\_SI

1: SI (billete >= PA Y SA>0) ENTONCES

vuelto <- billete-PA

i <- billete-vuelto

SA <- SA-1

ESCRIBIR (vuelto, i, SA)

FIN\_SI

2: SI (billete >= PE Y SE>0) ENTONCES

vuelto <- billete-PE

i <- billete-vuelto

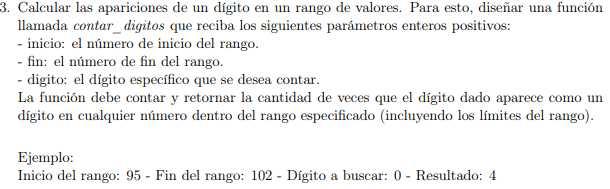
SE <- SE-1

ESCRIBIR (vuelto, i, SE)

FIN\_SI

FIN\_SEGÚN

A1.3 FIN\_MIENTRAS



Función contar\_digitos (inicio, fin, digito): enteros positivos -> enteros positivos

Hacer fin veces (i: inicio…fin)

i <- i+1

SEGÚN i:

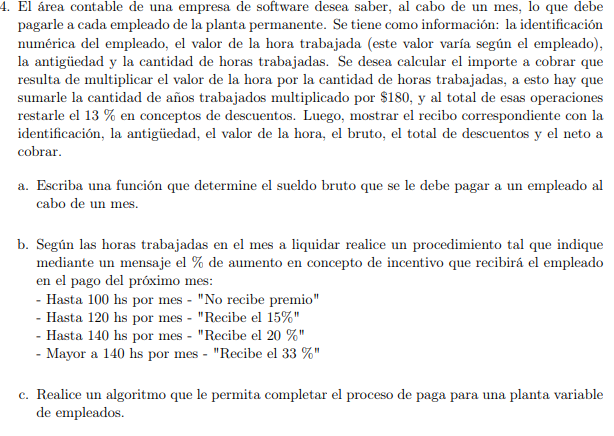
digito: cont <- cont +1

FIN\_SEGÚN

Fin\_Hacer

RETORNA cont

Fin\_Función



ALGORITMO sueldos

ENTRADA: ine: entero positivo, empleados: entero positivo, vht: real, antigüedad, cht: real

SALIDA: bruto: reales, descuentos: reales, cobra: reales

Var. Aux: i: entero positivo

CONSTANTES: RESTAR: 0.87

A1. LEER (empleados)

A2. Calcular\_recibo

A3. Parar

A2. Calcular\_recibo

A2.1 HACER empleados VECES (i:1…empleados)

A2.2 LEER (ine, vht, cht, antigüedad)

A2.3 bruto <- Resultado

A2.4 reciboEmpleado (bruto, antigüedad)

A2.5 Fin\_Hacer

Función bruto (vht, cht, antigüedad): entero, real, real -> real

Resultado <- (vht\*cht) + (antigüedad\*180)

RETORNA Resultado

Fin\_función

Procedimiento reciboEmpleado (cht, bruto, antigüedad)

cobra <- bruto\*RESTAR

descuentos <- bruto-cobra

ESCRIBIR(cobra, descuentos, bruto)

SI (cht<=100) ENTONCES

ESCRIBIR (“No recibe premio”)

SINO SI (cht<=120) Entonces

ESCRIBIR(“Recibe el 15%”)

SINO SI (cht<=140) Entonces

ESCRIBIR(“Recibe el 20%”)

SINO

ESCRIBIR(“Recibe el 33%”)

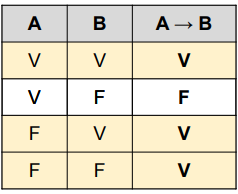
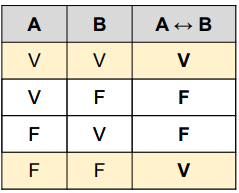
FIN\_SI

Fin\_Procedimiento

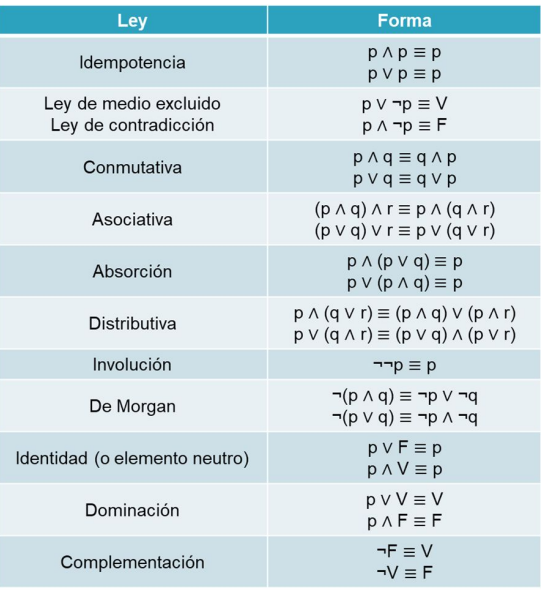
**\*FIN ALGORITMOS\***

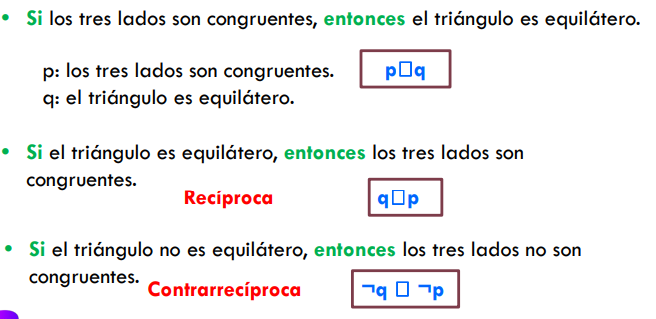
**\*LÓGICA COMPUESTA Y PROPOSICIONAL\***

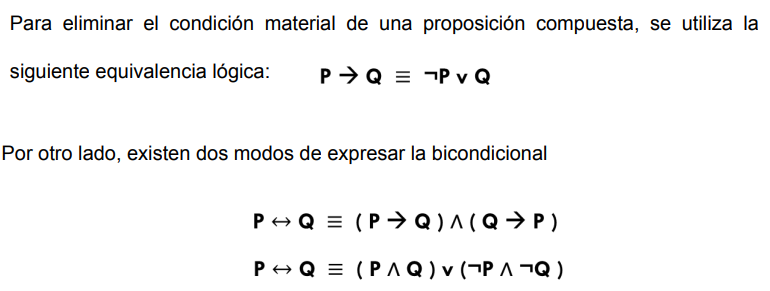
**TABLA DE VERDAD Y USO**

** **

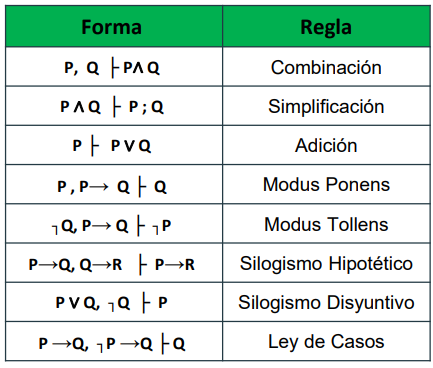
**LEYES LÓGICAS**

****

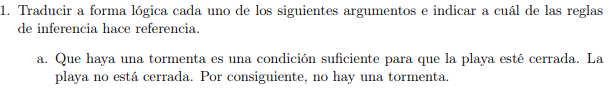
****

****

**REGLAS DE INFERENCIA**

****

**Si hay algo negado o algo normal se puee por equivalencia lógica dar un igual. Por ejm: r’ v s. Se puede hacer r=s (r ->s)**



P: hay una tormenta. q: la playa está cerrada.

pq (Modus Tollens)



P: estaré en forma. Q: me ejercito diariamente. R: voy a correr una maratón

qp, pr qr (Silogismo Hipotético)



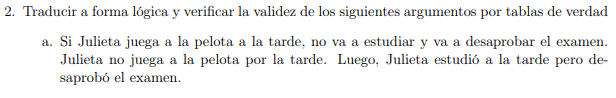
p: el cielo está despejado. Q: no habrá lluvia

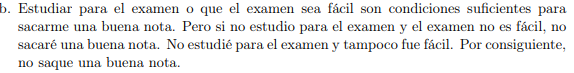
p, p (Modus Ponens)



P: Miguel recibirá un aumento. Q: Miguel recibirá un bono de fin de año.

P v q, q (Silogismo Disyuntivo)





P: estudio para el examen. Q: el examen es fácil. R: saco buena nota

[((p v q) r) ((

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p | q | r | (p v q) r | ( | [((p v q) r) (( | [((p v q) r) (( |
| V | V | V | V | V | V | V |
| V | V | F | F | V | F | F |
| V | F | V | V | V | V | V |
| F | V | V | V | V | V | V |
| F | F | V | V | F | F | F |
| V | F | F | F | V | F | F |
| F | V | F | F | V | F | F |
| F | F | F | V | V | V | V |



¬t; s → t; p → q; q → r; r → s ⊢ ¬p

1. ¬t premisa
2. s → t premisa
3. p → q premisa
4. q → r premisa
5. r → s premisa
6. p→r x3, 4 y Silogismo hipotético
7. p→s x6, 5 y Silogismo hipotético
8. p→t x7, 2 y Silogismo hipotético
9. ¬p x8, 1 y Modus Tollens

¬r ∧ (p → q); q → r ⊢ ¬p

1. ¬r ∧ (p → q) premisa
2. q → r premisa
3. ¬r; p → q x1 y Ley de simplificación
4. p → r x3, 2 y Silogismo hipotético
5. ¬p x4, 3 y Modus Tollens

(q ∧ r) → s; ¬s; p → (r ∧ q) ⊢ ¬p

1. (q ∧ r) → s premisa
2. ¬s premisa
3. p → (r ∧ q) premisa
4. q; r → s x1 y Ley de simplificación
5. p → r; q x 3 y Ley de simplificación
6. p → s; q x5, 4 y Silogismo hipotético
7. q x5, 4 y simplificación
8. ¬p x6, 2 y Modus Tollens
9. ¬p ∧ q x8, 7 y Ley de combinación
10. ¬p x9 y Ley de adición

p → q; ¬q ∨ s; p ∨ (r ∧ s) ⊢ s

1. p → q premisa
2. ¬q ∨ s premisa
3. p ∨ (r ∧ s) premisa
4. q → s x2 y Equivalencia lógica
5. p → s x4, 1 y Silogismo hipotético
6. (p v r) ∧ (p v s) x3 y Ley distributiva
7. p v s x6 y Ley de simplificación
8. ¬p → s x7 y Equivalencia lógica
9. S x8, 4 y Ley de casos

q → s; ¬(¬q ∨ p); ¬p → r ⊢ r ∧ s

1. q → s premisa
2. ¬(¬q ∨ p) premisa
3. ¬p → r premisa
4. ¬q → ¬p x2, Equivalencia lógica y Ley de Morgan
5. ¬q → r x4, 3 y Silogismo hipotético

p; r → q; p → (¬q ∨ ¬s); ¬(q ∧ s) → ¬(q ∨ r) ⊢ ¬r

1. P premisa
2. r → q premisa
3. p → (¬q ∨ ¬s) premisa
4. ¬(q ∧ s) → ¬(q ∨ r) premisa



p ∨ (q ∧ r); p → r ⊢ r

1. p ∨ (q ∧ r) premisa
2. p → r premisa
3. ¬r por hipótesis de lo absurdo
4. (p V q) ∧ (p V q) x1 y Ley distributiva
5. (p V q) x4 y Ley de simplificación