

## Capítulo 2 – Rios Nolasco Angelo Patrick

### 2.1.

```
IF x>=10 THEN
  DO
    x = x - 5
    IF x<50 EXIT
  ENDDO
ELSE
  IF x<5 THEN
    x = 5
  ELSE
    x = 7.5
ENDIF
```

### 2.2

```
DO
  i = i + 1
  IF z > 50 EXIT
  x = x + 5
  IF x > 5 THEN
    y = x
  ELSE
    y = 0
  ENDIF
  z = x + y
ENDDO
```

### 2.3

```
# algoritmo hecho con python
def algoritmoCartas():
    import random;
    tarjetas = [];

    # 1. Ingresar los valores para la concentración de un contaminante
    for i in range(0, 52):
        valueTarjeta = random.randrange(0, 100);
        tarjetas.append(valueTarjeta);

    # 2. Identificamos los datos que necesitamos calcular
    suma = 0;
    promedio = 0;
    max = 0;

    for i in tarjetas:
        # 3. Sumamos todos los valores que tenemos
        suma = suma + i;
        # 4. Calcular el valor máximo de todos los valores
        if(i>max):
            max = i;
        # 5. Calculamos el promedio dividiendo la suma entre la cantidad de valores
        promedio = suma / 52;
        # 6. Mostrar los valores obtenidos
        print(f"Suma {suma} -- Promedio {promedio} -- valor máximo {max}");
    algoritmoCartas();
```

2.3 En cada una de las tarjetas de un conjunto de cartas índice, se registra un valor para la concentración de un contaminante en un lago. Al final del conjunto, se coloca una carta marcada como "fin de los datos". Escriba un algoritmo para determinar la suma, el promedio y el máximo de dichos valores.



suma = a1+a2+...+a52

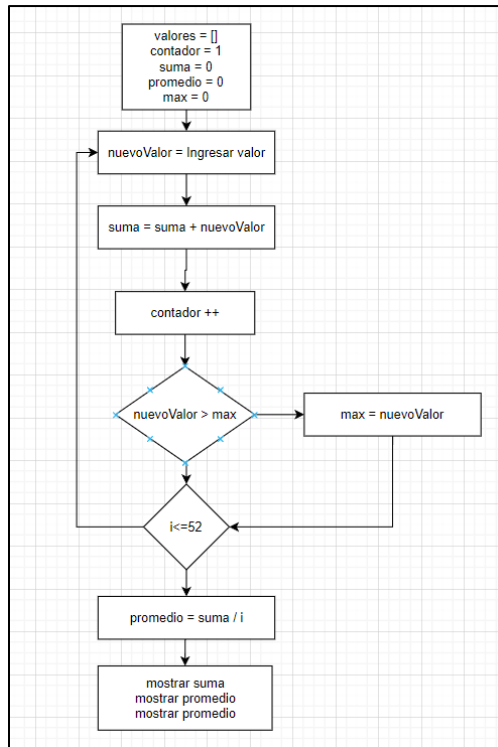
promedio = suma/52

max = a1

if valor > max  
max = valor

para calcular el max

## 2.4



## 2.5

```

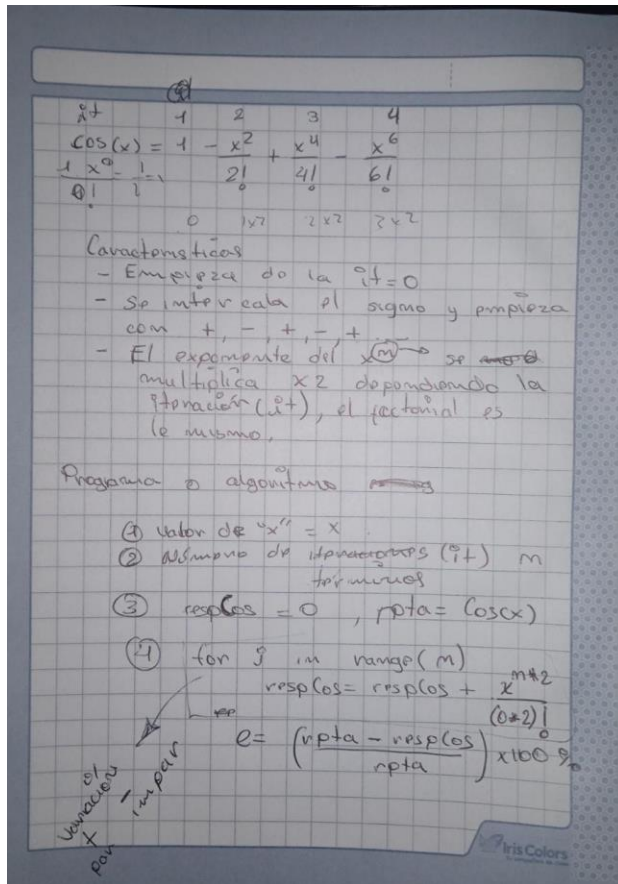
# ejercicio para calcular las raices de una ecuación cuadrática
def ejercicioCuadratica():
    import math;
    import cmath;
    # 1. Solicitar los valores al usuario
    a = float(input("Ingrese el valor de 'a': "));
    b = float(input("Ingrese el valor de 'b': "));
    c = float(input("Ingrese el valor de 'c': "));

    def calcularDiscriminante(a,b,c):
        return math.pow(b,2) - 4*a*c;

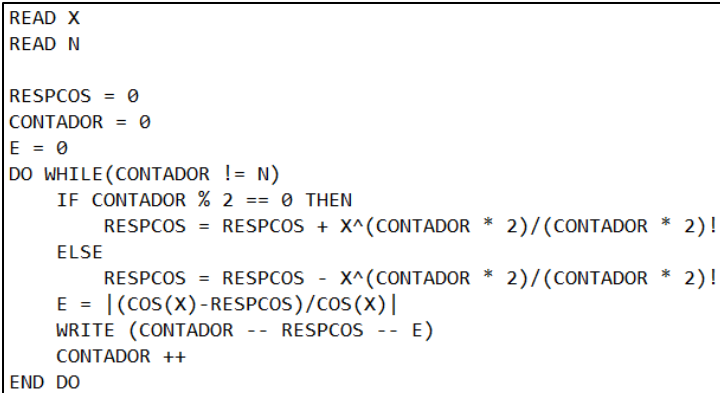
    discriminante = calcularDiscriminante(a,b,c);

    def calcularRaices(a,b,discriminante):
        if discriminante > 0:
            return print(f"x1 = {(-b+math.sqrt(discriminante))/(2*a)} ; x2 = {(-b-math.sqrt(discriminante))/(2*a)}");
        if discriminante == 0:
            return print(f"x(1,2) = {-b/(2*a)}");
        if discriminante < 0:
            return print(f"x1 = {(-b+cmath.sqrt(discriminante))/(2*a)} ; x2 = {(-b-cmath.sqrt(discriminante))/(2*a)}")
    try:
        calcularRaices(a,b,discriminante);
    except:
        print("Existe división entre cero");
    ejercicioCuadratica();
  
```

## 2.6



Paso 1: Guardar número del coseno ('x') y número de iteraciones ('n')  
 Paso 2: Definir la variable 'respCos' y el contador 'contador'  
 Paso 3: verificamos si el 'contador' es par o impar  
 Paso 4: Si el 'contador' es par entonces, respCos va ser igual a 'respCos' más  $(x^{(\text{contador}*2)})/(\text{contador}*2)!$   
 Paso 5: Si el 'contador' es impar entonces, respCos va ser igual a 'respCos' menos  $(x^{(\text{contador}*2)})/(\text{contador}*2)!$   
 Paso 6: El error en cada iteración es  $(\cos(x) - \text{respCos})/\cos(x)$   
 Paso 7: Mostrar los resultados (contador -- respCos -- error)  
 Paso 8: Aumentamos el 'contador' en 1  
 Paso 9: Si contador no es igual a 'n' entonces, volvemos al paso 3º  
 Paso 10: Fin del programa



```
def calcularSerieCoseno():
    import math;
    from decimal import Decimal;

    # 1. Solicitar datos como el valor de x y el número de términos
    x = float(input("Cos(x) -> ingrese el valor de 'x': "));
    n = int(input("Ingrese el número de términos: "));

    # 2. Calculamos el oficial del cos(x)
    rpt_a = math.cos(x);
    print(f"El valor de cos({x}) = {round(Decimal(rpt_a), 6)}");

    # 3. Calculamos los valores de las iteraciones
    respCos = 0;
    print("it\t\t\t\t\t itCos\t\t\t\t\t error");
    for i in range(n):
        if(i%2 == 0):
            respCos = respCos + math.pow(x,i*2)/math.factorial(i*2);
        else:
            respCos = respCos - (math.pow(x,i*2))/(math.factorial(i*2));

    # 4. Calculamos el error
    e = abs((rpt_a - respCos)/rpt_a);

    # 5. imprimir los valores resueltos
    print(f"{i}\t\t\t\t\t {round(Decimal(respCos),6)}\t\t\t\t\t {round(Decimal(e),6)*100} %");
calcularSerieCoseno();
```

2.9

```

# pseudocódigo hecho en pseint
function value<-solicitarNotas(cantNotas)
    suma<-0;
    Para i<-0 Hasta cantNotas Con Paso 1 Hacer
        Escribir "Digite la nota ",(i+1);
        Leer nota;
        suma<-suma + nota;
    FinPara
    value<-suma / cantNotas;
FinFunction

Proceso Principal
    Escribir "Nombre del curso: ";
    Leer nombreCurso;
    Escribir "Clave de curso: ";
    Leer ClaveCurso;

    Escribir "Peso de los cuestionarios: ";
    Leer c;
    Escribir "Peso de las tareas: ";
    Leer t;

    pc<-0;
    pt<-0;
    nFinal<-0;
    promedioFinal<-0;
    cantNotas<-0;

    Escribir "Cant. de notas de los cuestionarios: ";
    Leer cantNotas;
    Leer pc<-solicitarNotas(cantNotas);
    Escribir "Cant. de notas de las tareas: ";
    Leer cantNotas;
    Leer pt<-solicitarNotas(cantNotas);

    Escribir "El curso tiene una nota final? si(1) - no(2): ";
    Leer pregunta;

    Si pregunta == 1 Entonces
        Escribir "Peso de la nota final: ";
        Leer ef;
        Escribir "Nota final: ";
        Leer nFinal;
        promedioFinal<-((c*pc + t*pt + ef*nFinal)/(c + t + ef))*100;
    SiNo
        promedioFinal<-((c*pc + t*pt)/(c + t))*100;
    Fin Si
    Escribir "Clave: ",claveCurso,"Curso: ",nombreCurso,"Promedio Final: ",promedioFinal;
FinProceso

```

```

def ejercicioCalificaciones():
    def solicitarNotas(cantNotas):
        suma = 0;
        for i in range(cantNotas):
            nota = int(input(f"Nota {i+1}: "));
            suma = suma + nota;
        return suma / cantNotas;

    nombreCurso = input("Ingrese el nombre del curso: ");
    claveCurso = input("Ingrese la clave del curso: ");
    c = int(input("Peso de la nota de los cuestionarios: "));
    t = int(input("Peso de la nota de los tareas: "));

    pc = 0;
    pt = 0;
    nFinal = 0;
    promedioFinal = 0;

    cantNotas = int(input("Cant. de notas de los cuestionarios: "));
    pc = solicitarNotas(cantNotas);

    cantNotas = int(input("Cant. de notas de la tarea: "));
    pt = solicitarNotas(cantNotas);

    pregunta = int(input("El curso tiene una nota final? si(1) - no(2): "));
    if pregunta == 1:
        ef = int(input("Peso de la nota final: "));
        nFinal = int(input("Ingrese la nota final del curso: "));
        promedioFinal = ((c*pc + t*pt + ef*nFinal)/(c + t + ef))*100;
    else:
        promedioFinal = ((c*pc + t*pt)/(c + t))*100;
    print(f"Clave: {claveCurso} -- Curso: {nombreCurso} -- Calificación Promedio: {promedioFinal}%");

```

2.10

```
READ A
TOL = 0
IF A>0 THEN
    DO WHILE(TRUE)
        Y = (X+(A/X))/2;
        E = |(Y-X)/Y|;
        X = Y;
        IF E<TOL THEN
            BREAK
        WRITE ("LA RAIZ APROXIMA ES: ",X)
    END DO
ELSE
    WRITE ("LA RAIZ CUADRADA ES 0");
```

```
def aproximacionRaiz():
    from decimal import Decimal
    a = float(input("Ingresa el valor de la raiz: "));
    tol = 0.00001;
    if a > 0:
        x = a/2;
        while True:
            y = (x+(a/x))/2;
            e = round(Decimal(abs((y-x)/y)),6);
            x = y;
            print(f"Aproximación de sqrt({a}) = {x} con un e = {e}");
            if(e<tol):
                break;
    else:
        print("Raiz cuadrada = 0");
    aproximacionRaiz();
```

## 2.11

# ejercicio 2.11 calcular el valor futuro

```
def calcularValorFuturo():
    p = float(input("Ingrese la inversión inicial: "));
    tasa = float(input("Ingrese la tasa de interés (decimal): "));
    n = int(input("Número de años: "));

    print(f"Periodo          Valor futuro");
    for i in range(n):
        f = p*math.pow((1+tasa),i+1);
        print(f"{i+1}          {round(Decimal(f),4)} $");
    calcularValorFuturo();|
```

## 2.12

```
def calcularPagoAnual():
    from decimal import Decimal;
    p = float(input("Ingrese el prestamo realizado: "));
    ti = float(input("Ingrese la tasa de interés (decimal): "));
    n = int(input("Ingrese la cantidad de pagos: "));
    a = (p*ti*math.pow((1+ti), n))/(math.pow((1+ti),n)-1);
    print("n          capital          interes          amotiz          cuotaAnual");
    interesTotal = 0;
    amotizacionTotal = 0;
    for i in range(n):
        itc = p * ti;
        interesTotal = interesTotal + itc;
        amotizacion = a - itc;
        amotizacionTotal = amotizacionTotal + amotizacion;
        print(f"{i+1} {round(Decimal(p),4)} {round(Decimal(itc),4)} {round(Decimal(amotizacion),4)} {round(Decimal(a),4)}");
        p = p - amotizacion;
    print(f"Final-> {round(Decimal(p),4)} {round(Decimal(interresTotal),4)} {round(Decimal(amotizacionTotal),4)} {round(Decimal(a*n),4)}");
    calcularPagoAnual();
```

## 2.13



2.14

```
def paracaidas():
    ca = float(input("Ingrese el coeficiente de rozamiento: "));
    m = float(input("Ingrese la masa del paracaidas: "));

    #número de iteraciones
    contador = int(input("Cuántos recorridos desea?: "))

    #velocidad inicial
    v = 0;

    #tamaño de pasos
    pasos = 1;

    #gravedad
    g = 9.8;
    t1 = 0;
    t2 = 0;

    print("t(s)    v(m/s)");
    for i in range(contador+1):
        v = v+(g-(ca*v)/(m))*(t2-t1);

        # imprimir valores
        print(f"{t2}          {round(Decimal(v),5)}");
        t1 = t2;
        t2 = t2 + pasos;
```

2.15

```
# 2.15 ejercicio de la burbuja

def metodoBurbuja():
    import random;
    numeros = [];

    # insetar números aleatorios
    for i in range(20):
        numeros.append(random.randint(0, 100));

    aux = 0;

    # método totalmente ineficiente del ordenamiento de la burbuja
    for i in range(len(numeros)):
        for j in range(len(numeros)-1):
            if numeros[j] > numeros[j+1]:
                aux = numeros[j];
                numeros[j] = numeros[j+1];
                numeros[j+1] = aux;
```

2.16

2.17

```
# ejercicio 2.17 calcular polares
def calcularPolares():
    # 1. Saber la coordenada

    while(True):
        x = float(input("Ingrese el valor de x: "));
        y = float(input("Ingrese el valor y: "));

        # 2. radio (r)
        r = math.sqrt(math.pow(x,2) + math.pow(y,2));

        angulo = 0;

        if x<0 and y>0:
            angulo = math.degrees(math.atan(y/x)) + 180;
        if x<0 and y<0:
            angulo = math.degrees(math.atan(y/x)) - 180;
        if x<0 and y==0:
            angulo = 180;
        if x==0 and y>0:
            angulo = 90;
        if x==0 and y<0:
            angulo = -90;
        if x==0 and y==0:
            angulo = 0;
        if x>0:
            angulo = math.degrees(math.atan(y/x));

        if angulo < 0:
            angulo = angulo + 360;
        print(f"r = {r} ; angulo={angulo}")
```

2.18

```
# ejercicio 2.18 condicionales
def procedimientosCondicionales():
    num = int(input("Ingrese un número: "));

    if 90<=num<=100:
        print("A");
    if 80<=num<90:
        print("B");
    if 70<=num<80:
        print("C");
    if 60<=num<70:
        print("D");
    if num<60:
        print("F");
    if num>100:
        print("No tiene clasificación numérica.");
```

## 2.19

a)

```
# ejercicio 2.19
def ejercicioFactorial():
    number = int(input("Número: "))
    factorial = 1;
    for i in range(1, number+1):
        factorial = factorial * i;
    print(f"Factorial es: {factorial}")
ejercicioFactorial();
```

b)

```
# ejercicio 2.19 b
def menorValorVector():
    numbers = [7,2,1,51,9,5,17,1];
    for i in range(len(numbers)):
        pos = i;
        aux = numbers[i];
        while ((pos>0) and (numbers[pos-1] > aux)):
            numbers[pos] = numbers[pos-1];
            pos -= 1;
        numbers[pos] = aux;
    print(numbers[0]);
menorValorVector();
```

c)

```
# ejercicio 2.19 c
def promedioVector():
    numbers = [7,2,1,51,9,5,17,1];
    suma = 0;

    for i in numbers:
        suma = suma + i;
    print(f"promedio es: {suma/(len(numbers))}")
promedioVector();
```

## 2.20

a)

```
# ejercicio 2.20 a
def sumaCuadradosMatriz():
    matriz = [[1,2,3],[2,3,5]];
    suma = 0;
    for i in range(len(matriz)):
        for j in range(len(matriz[i])):
            suma = suma + matriz[i][j]**2;
    print(f"la suma es: {suma**(0.5)}")
sumaCuadradosMatriz();
```

b)

```
# ejercicio 2.20 b

def normalizarMatriz():
    matriz = [[1,2,-3],[-2,3,-5]];

    for i in range(len(matriz)):
        maxValue = max(map((lambda num: num*-1),matriz[i]));
        for j in range(len(matriz[i])):
            matriz[i][j] = matriz[i][j] / maxValue;
    print(matriz);
normalizarMatriz();
```