**FUNDAMENTOS COMPUTACIONALES**

**Ejercicio 1**

1. Análisis

Generar Matriz con valores

Mostrar valores pares

Mostrar valores impares

Mostrar procedimiento

1. Pseudocodigo

Algoritmo ValoresParesEImpares

Definir matriz1, matriz2, matriz3, valorfil, valorcolum Como Entero

Definir prom Como Real

valorfil = 3

valorcolum = 6

Dimension matriz1[valorfil, valorcolum]

Dimension matriz2[valorfil, valorcolum]

Dimension matriz3[valorfil, valorcolum]

llenarMatriz(matriz1, valorfil, valorcolum)

Escribir "Matriz Generada:"

presentarMatriz(matriz1, valorfil, valorcolum)

llenarPares(matriz1, matriz2, valorfil, valorcolum)

Escribir "Pares de la matriz:"

presentarMatriz(matriz2, valorfil, valorcolum)

llenarImpares(matriz1, matriz3, valorfil, valorcolum)

Escribir "Impares de la matriz:"

presentarMatriz(matriz3, valorfil, valorcolum)

presentarPromedio(matriz1, valorfil, valorcolum, promedio)

Escribir "El promedio de los valores de la matriz es: ", promedio

FinAlgoritmo

SubProceso llenarMatriz (matriz1, valorfil, valorcolum)

Para i<-0 Hasta valorfil-1 Con Paso 1 Hacer

Para j<-0 Hasta valorcolum-1 Con Paso 1 Hacer

matriz1[i,j] = Aleatorio(1,100)

Fin Para

Fin Para

FinSubProceso

Funcion presentarMatriz (matriz1, valorfil, valorcolum)

Para i<-0 Hasta valorfil-1 Con Paso 1 Hacer

Para j<-0 Hasta valorcolum-1 Con Paso 1 Hacer

Escribir matriz1[i,j], " | " Sin Saltar

Fin Para

Escribir " "

Fin Para

FinFuncion

Funcion llenarPares (matriz1, matriz2, valorfil, valorcolum)

Para i<-0 Hasta valorfil-1 Con Paso 1 Hacer

Para j<-0 Hasta valorcolum-1 Con Paso 1 Hacer

Si (matriz1[i,j] MOD 2) == 0 Entonces

matriz2[i,j] = matriz1[i,j]

SiNo

matriz2[i,j] = 0

Fin Si

Fin Para

Fin Para

FinFuncion

Funcion llenarImpares (matriz1, matriz3, valorfil, valorcolum)

Para i<-0 Hasta valorfil-1 Con Paso 1 Hacer

Para j<-0 Hasta valorcolum-1 Con Paso 1 Hacer

Si (matriz1[i,j] MOD 2) <> 0 Entonces

matriz3[i,j] = matriz1[i,j]

SiNo

matriz3[i,j] = 0

Fin Si

Fin Para

Fin Para

FinFuncion

Funcion presentarPromedio (matriz1, valorfil, valorcolum, promedio Por Referencia)

Definir sum Como Entero

Para i<-0 Hasta valorfil-1 Con Paso 1 Hacer

Para j<-0 Hasta valorcolum-1 Con Paso 1 Hacer

sum = sum + matriz1[i,j]

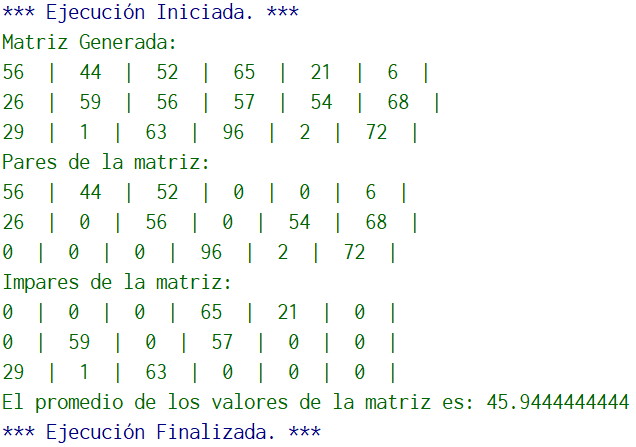
Fin Para

Fin Para

promedio = sum/(valorfil\*valorcolum)

FinFuncion

1. Prueba de Escritorio



**Ejercicio 2**

1. Análisis

Generar solución de 3 procedimientos

Calcular área del cuadrado

Calcular área del triangulo

Calcular área del rectángulo

Mostrar áreas de cada uno

1. Pseudocodigo

Algoritmo AreaCua\_Tri\_Rec

Definir opcNum Como Entero

Definir lado, baseTri, alturaTri, baseRec, alturaRec, areaTotal Como Real

areaTotal=0

Escribir "Escoga la opcion para calcular: "

Escribir "1. Area del Cuadrado"

Escribir "2. Area del Triangulo"

Escribir "3. Area del Rectangulo"

Leer opcNum

Si (opcNum ==1)

Escribir "Escriba el lado del cuadrado: "

Leer lado

areaTotal<-areaCuadrado(lado)

Escribir "El area del cuadrado es: ", areaTotal

SiNo

Si (opcNum ==2)

Escribir "Ingrese la base del triangulo: "

Leer baseTri

Escribir "Ingrese la altura del triangulo: "

Leer alturaTri

areaTotal<-areaTriangulo(baseTri, alturaTri)

Escribir "El area de un triangulo es: ", areaTotal

SiNo

Si(opcNum ==3)

Escribir "Ingrese la base del rectangulo: "

Leer baseRec

Escribir "Ingrese la altura del rectangulo: "

Leer alturaRec

areaTotal<-areaRectangulo(baseRec, alturaRec)

Escribir "El area del rectangulo es: ", areaTotal

SiNo

FinSi

FinSi

FinSi

FinAlgoritmo

SubProceso areaTotal<-areaCuadrado(lado)

areaTotal = lado \* lado

FinSubProceso

SubProceso areaTotal<-areaTriangulo(baseTri, alturaTri)

areaTotal= (baseTri \* alturaTri) / 2

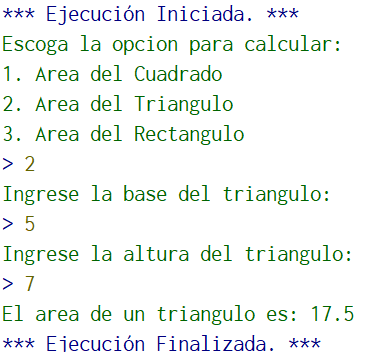
FinSubProceso

SubProceso areaTotal<-areaRectangulo(baseRec, alturaRec)

areaTotal = baseRec \* alturaRec

FinSubProceso

1. Prueba de Escritorio



**Ejercicio 3**

1. Analisis

Generar función de 4 parametros

Mostrar si es promedio regular

Mostrar si es promedio bueno

Mostrar si es promedio muy bueno

Mostrar si es promedio sobresaliente

1. Pseudocodigo

Algoritmo Promedios

Definir param1, param2, param3, param4, prom Como Real

Definir cualit Como Cadena

Escribir "Ingrese la nota del parametro 1:"

Leer param1

Escribir "Ingrese la nota del parametro 2:"

Leer param2

Escribir "Ingrese la nota del parametro 3:"

Leer param3

Escribir "Ingrese la nota del parametro 4:"

Leer param4

Escribir "El promedio cualitativo es: ", califiCuali(param1,param2,param3,param4)

FinAlgoritmo

Funcion cualit<-califiCuali(param1,param2,param3,param4)

prom = (param1 + param2 + param3 + param4) / 4

Si(prom <= 5)

cualit = "Regular"

SiNo

Si(prom <= 8)

cualit = "Bueno"

SiNo

Si(prom <= 9)

cualit = "Muy bueno"

SiNo

cualit = "Sobresaliente"

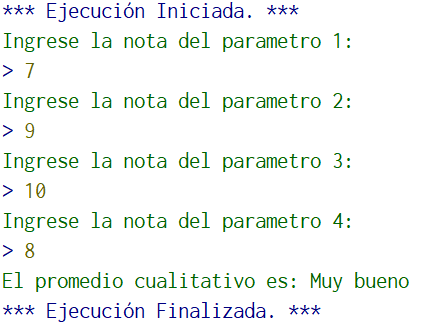
FinSi

FinSi

FinSi

FinFuncion

1. Prueba de Escritorio



**Ejercicio 4**

1. Analisis

Generar procedimiento para planilla de luz y predio de un inmueble

Ingresar nombre del usuario

Ingresar cedula de usuario

Mostrar cuanto debe pagar de planilla de luz

Mostrar cuanto debe pagar del predio de un inmueble

1. Pseudocodigo

Algoritmo PlanillaLuz\_PredioInmueble

Definir opc Como Entero

Definir valorKW, numKV, valorInmueb Como Real

Definir cedula, nombCli Como Cadena

Escribir "Ingrese el nombre del cliente: "

Leer nombCli

Escribir "Ingrese la cedula del cliente: "

Leer cedula

Escribir "Ingrese la opcion a consultar: "

Escribir "1. Planilla de luz"

Escribir "2. Predio de un bien inmueble"

Leer opc

Si (opc == 1 )

Escribir "Ingresar el valor del Kilovatio: "

Leer valorKW

Escribir "Ingrese el numero de Kilovatios gastados al mes: "

Leer numKV

valorPlanilla<-planillaLuz(valorKW, numKV)

Escribir "Reporte"

Escribir "Cliente ", nombCli, " con cedula ", cedula, " debe pagar el valor de $", valorPlanilla

SiNo

Si (opc == 2)

Escribir "Ingresar el valor del Inmueble: "

Leer valorInmueb

predio<-calcPredio(valorInmueb)

Escribir "Reporte"

Escribir "Cliente ", nombCli, " con cedula ", cedula, " tiene un inmueble valorado en $", valorInmueb," y tiene que pagar de predio $", predio

FinSi

FinSi

FinAlgoritmo

SubProceso valorPlanilla<-planillaLuz(valorKW, numKV)

Definir valorPlanilla Como Real

valorPlanilla = valorKW \* numKV

FinSubProceso

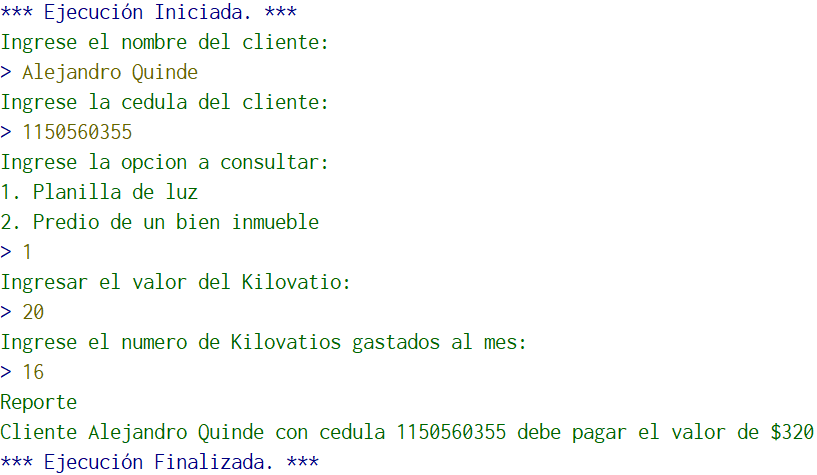
SubProceso predio<-calcPredio(valorInmueb)

Definir predio Como Real

predio = valorInmueb \* 0.02

FinSubProceso

1. Prueba de Escritorio



**Ejercicio 5**

1. Analisis

Generar función que devuleva suma

Generar función que devuelva resta

Generar función que devuelva multiplicacion

Motras suma, resta y multiplicación

1. Pseudocodigo

Algoritmo OperacionesMatriz

Definir valorfil,valorcolum Como Entero

Escribir "Ingrese la cantidad de filas y columnas de la matriz:"

Leer valorfil

Leer valorcolum

Dimension matrizA[valorfil, valorcolum]

llenarMatriz(matrizA, valorfil, valorcolum)

Escribir "Matriz Generada:"

presentarMatriz(matrizA, valorfil, valorcolum)

Definir suma, resta, multiplicacion Como Entero

suma = sumarMatriz(matrizA, valorfil, valorcolum)

resta = restarMatriz(matrizA, valorfil, valorcolum)

multiplicacion = multiplicarMatriz(matrizA, valorfil, valorcolum)

Escribir "La suma de los elementos de la matriz es:", suma

Escribir "La resta de los elementos de la matriz es:", resta

Escribir "La multiplicación de los elementos de la matriz es:", multiplicacion

FinAlgoritmo

SubProceso llenarMatriz(matriz, valorfil, valorcolum)

Para i <- 0 Hasta valorfil-1 Con Paso 1 Hacer

Para j <- 0 Hasta valorcolum-1 Con Paso 1 Hacer

matriz[i, j] = Aleatorio(1, 9)

FinPara

FinPara

FinSubProceso

Funcion presentarMatriz(matriz, valorfil, valorcolum)

Para i <- 0 Hasta valorfil-1 Con Paso 1 Hacer

Para j <- 0 Hasta valorcolum-1 Con Paso 1 Hacer

Escribir matriz[i, j], " | " Sin Saltar

FinPara

Escribir " "

FinPara

FinFuncion

Funcion suma<-sumarMatriz(matriz, valorfil, valorcolum)

Definir suma Como Entero

suma = 0

Para i <- 0 Hasta valorfil-1 Con Paso 1 Hacer

Para j <- 0 Hasta valorcolum-1 Con Paso 1 Hacer

suma = suma + matriz[i, j]

FinPara

FinPara

FinFuncion

Funcion resta<-restarMatriz(matriz, valorfil, valorcolum)

Definir resta Como Entero

Para i <- 0 Hasta valorfil-1 Con Paso 1 Hacer

Para j <- 0 Hasta valorcolum-1 Con Paso 1 Hacer

Si i = 0 Y j = 0 Entonces

resta = matriz[i, j]

Sino

resta = resta - matriz[i, j]

FinSi

FinPara

FinPara

FinFuncion

Funcion multiplicacion<-multiplicarMatriz(matriz, valorfil, valorcolum)

Definir multiplicacion Como Entero

multiplicacion = 1

Para i <- 0 Hasta valorfil-1 Con Paso 1 Hacer

Para j <- 0 Hasta valorcolum-1 Con Paso 1 Hacer

multiplicacion = multiplicacion \* matriz[i, j]

FinPara

FinPara

FinFuncion

1. Prueba de Escritorio

