PROYECTO

FUNDAMENTOS COMPUTACIONALES

Integrantes: Eimer Armijos, Alejandro Quinde

EJERCICIO 1

1. ANALISIS

Entrada Proceso Salida

Numero de Estudiantes Generar nombre Porcentaje Aprobado

Generar número de cedula Porcentaje Reprobado

Generar notas Datos de Estudiantes

Sacar porcentajes Nota final

Comparar si aprobó o no

Sacar promedio

1. DISEÑO EN PSEUDOCODIGO

Inicio

caracter <- nombre

caracter <- cedula

real <- adc=azar(10)

real <- ape=azar(10)

real <- aa=azar(10)

i <- 0

Escribir 'ingrese el nunero de estudiantes'

Leer numEstudiantes

Para i<-0 Hasta numEstudiantes Con Paso 1 Hacer

nombre <- 'estudiante'

cedula <- azar(1000000000)

Escribir 'estudiante', nombre

Escribir 'cedula', cedula

adc <- adc\*0.35

Escribir 'ADC', adc

ape <- ape\*0.35

Escribir 'APE', ape

aa <- aa\*0.30

Escribir 'AA', aa

nota <- aa+adc+ape

Escribir 'todas las notas: ', nota

Si (nota>7) Entonces

Escribir 'Aprobado'

aprobados <- aprobados+1

SiNo

Escribir 'Estudiante debera realizar un examen de recuperacion'

notaRecuperada <- (notaExamnen\*0.35)+(nota\*0.6)

Escribir 'Nota del examen de recuperacion: ', (notaExamen)

FinSi

Si notaRecuperada<7 Entonces

Escribir 'Reprobado totalmente'

Escribir 'nota: ', nota

SiNo

Escribir 'Aprobado'

aprobados <- aprobados+1

FinSi

FinPara

Escribir 'Estadisticas:'

Escribir 'Aprobados: ', aprobados

Escribir 'Reprobados: ', reprobados

porAprobados <- (aprobados/numEstudiantes)\*100

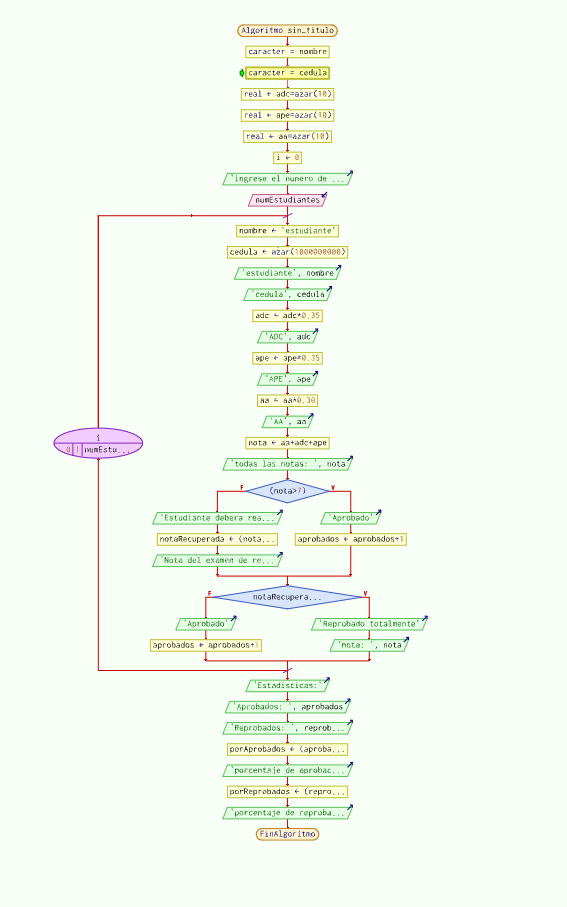
Escribir 'porcentaje de aprobacion: ', porAprobados, '%'

porReprobados <- (reprobados/numEstudiantes)\*100

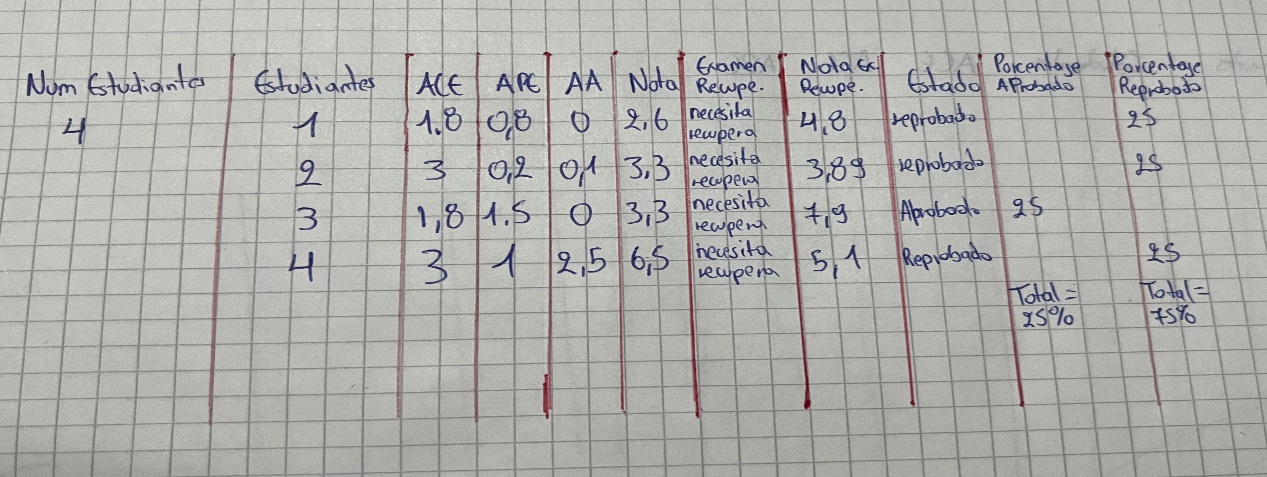
Escribir 'porcentaje de reprobacion: ', porReprobados, '%'

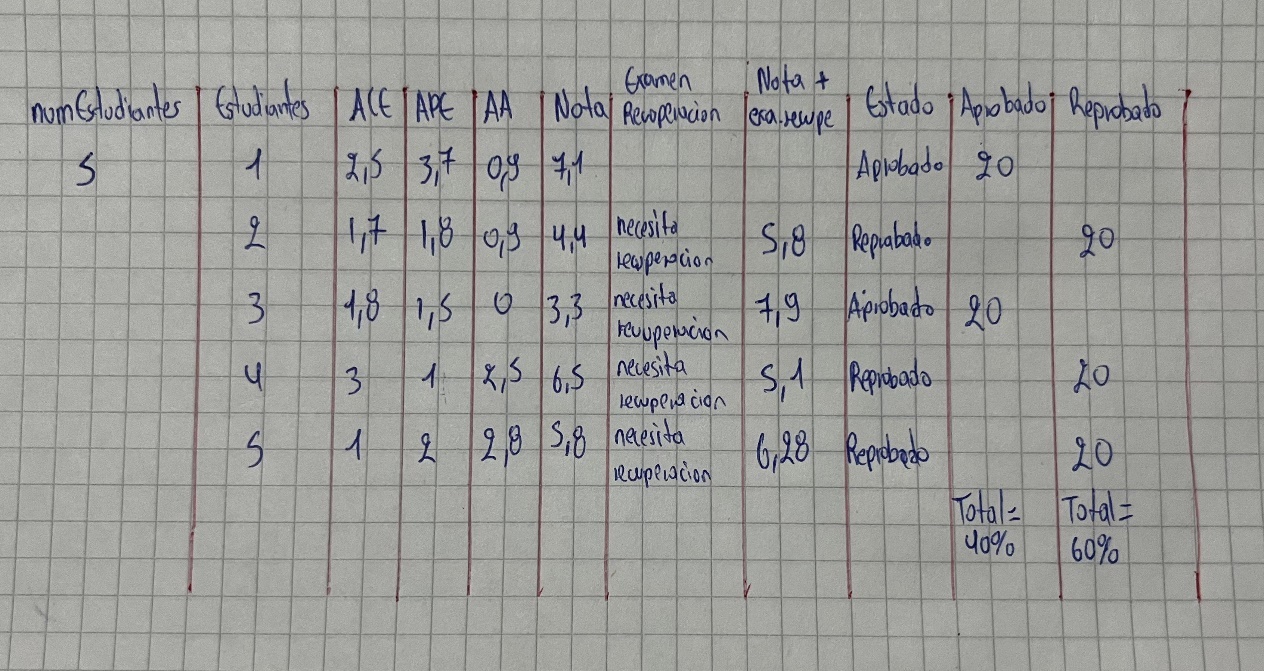
FinAlgoritmo

1. FLUJOGRAMA



1. PRUEBA DE ESCRITORIO





EJERCICIO 2

1. ANALISIS

|  |  |  |
| --- | --- | --- |
| datos de entrada | proceso | salida |
| Ingresar numero | Generar un numero al azar  Comprobar los números  Dar pistas | Lo ha logrado  No lo logro  Se termino  Sin intentos |

1. DISEÑO EN PSEUDOCODIGO

Inicio

Proceso AdivinarNumero

Definir teclado, intentos, numAdiv, intentosFalt, pista como Entero

Definir pistaInput, nuevoIntento como Caracter

teclado <- 0

numAdiv <- Aleatorio(1, 100)

intentosFalt <- 10

Mientras intentosFalt > 0 Hacer

Escribir("Ingresa un número del 1 al 100: ")

Leer intentos

Si intentos = numAdiv Entonces

Escribir("Bien, has adivinado el número")

CortarProceso

Sino

intentosFalt <- intentosFalt - 1

Escribir("Te quedan ", intentosFalt, " intentos.")

Escribir("")

Si intentosFalt > 0 Entonces

Escribir("Tienes opción a una pista (si/no): ")

Leer pistaInput

pistaInput <- Minusculas(pistaInput)

Si pistaInput = "si" Entonces

Escribir("Ingresa el número de la pista que deseas: ")

Escribir("Seleccione (1). Si quiere saber si el número es mayor o menor")

Escribir("Seleccione (2). Si quiere saber si el número es par o impar")

Escribir("Seleccione (3). Si quiere saber si el número es primo")

Escribir("Seleccione (4). Si quiere saber si el número es múltiplo de algún número")

Escribir("")

Leer pista

Segun pista Hacer

Caso 1:

Si numAdiv < intentos Entonces

Escribir("- PISTA: Intenta con un número menor")

Sino

Escribir("- PISTA: Intenta con un número mayor")

FinSi

Caso 2:

Si numAdiv MOD 2 = 0 Entonces

Escribir("- PISTA: El número es par")

Sino

Escribir("- PISTA: El número es impar")

FinSi

Caso 3:

Si Primo(numAdiv) Entonces

Escribir("- PISTA: El número es un número primo")

Sino

Escribir("- PISTA: El número no es un número primo")

FinSi

Caso 4:

Escribir("Ingrese el posible divisor: ")

Leer div

Si Multiplo(numAdiv, div) Entonces

Escribir(numAdiv, " es múltiplo de ", div)

Sino

Escribir("No es múltiplo de ", div)

FinSi

FinSegun

intentosFalt <- intentosFalt - 1

Escribir("Te quedan: ", intentosFalt, " intentos")

FinSi

FinSi

FinSi

FinMientras

Si intentosFalt = 0 Entonces

Escribir("Te has quedado sin intentos. El número es: ", numAdiv)

FinSi

Escribir("¿Quieres jugar de nuevo? (si/no): ")

Leer nuevoIntento

nuevoIntento <- Minusculas(nuevoIntento)

Si nuevoIntento = "si" Entonces

FinSi

FinProceso

Funcion Multiplo(intento, div)

Devolver = intento MOD div = 0

FinFuncion

Funcion Primo(intento)

Si intento <= 1 Entonces

Devolver Falso

FinSi

Para i <- 2 Hasta RaizCuadradaDe(intento) Con Paso 1 Hacer

Si intento MOD i = 0 Entonces

Devolver = Falso

FinSi

FinPara

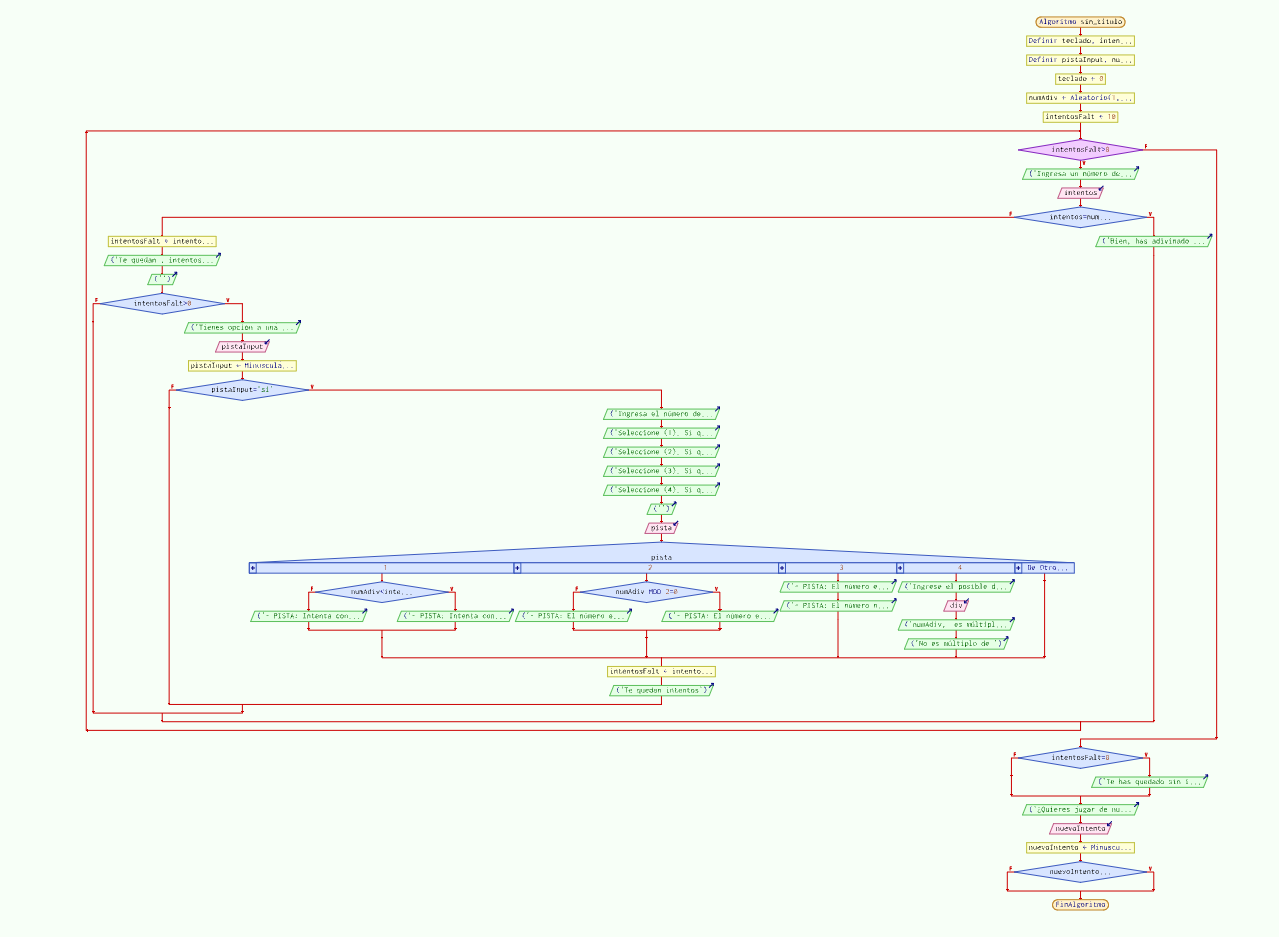
Devolver = Verdadero

FinFuncion

FinProceso

Fin

1. FLUJOGRAMA



1. PRUEBA DE ESCRITORIO

|  |  |  |  |
| --- | --- | --- | --- |
| intento | numero usuario | pista | numero adivinado |
| 1 | 12 | es primo |  |
| 2 | 1 | es mayor |  |
| 3 | 2 | es multpli de 3 |  |
| 4 | 3 | | ¡Felicidades Has adivinado el número! |

|  |  |  |  |
| --- | --- | --- | --- |
| intento | numero usuario | pista | numero adivinado |
| 1 | 34 | es par |  |
| 2 | 2 | es mayor |  |
| 3 | 4 | es multiplo de 2 |  |
| 4 | 6 | es mayor |  |
| 5 | 8 |  | Lo siento, has agotado tus intentos. |