



# Proyecto Final.

UDA: Fundamentos de los Sistemas de la Información

Docente: Ing. Alfredo Medina García

Alumnos: Leonardo Daniel Aviña Neri

Alan Castro Martínez

Jorge Luis Ramírez Ramírez

Carrera: Licenciatura En Ingeniería De Datos E Inteligencia Artificial (LIDIA)

Fecha de entrega: 06/12/2024



Índice

Problema 1..... 2

    Enunciado..... 2

    Pseudocodigo ..... 2

    Codigo ..... 3

Problema 2..... 4

    Enunciado..... 4

    Pseudocódigo ..... 4

    Código ..... 5

Problema 3..... 7

    Enunciado..... 7

    Pseudocódigo ..... 7

    Código ..... 9



# Problema 1

## Enunciado

Desarrollar un juego en donde el usuario debe adivinar un numero entre 0 y 100 generado aleatoriamente por el programa

El programa debe generar un numero aleatorio diferente cada vez que se ejecuta el código.

Debe decir al usuario en cada intento si el numero ingresado es mayor o menor que el número correcto hasta que adivine el número.

Al final, cuando el usuario adivine el número, debe aparecer un mensaje que le diga que gano y debe mostrar cuantos intentos hizo el usuario para adivinar el número.

## Pseudocodigo

INICIO

```
Inicializar intentos = 0, num = -1
aleatorio = número aleatorio entre 0 y 100
do
    print "Adivina el número: "
    leer num
    intentos=intentos+1

    If num > aleatorio:
        print "El número debe ser menor"
    Else if num < aleatorio:
        print "El número debe ser mayor"
    Fin If
while num != aleatorio
print "Ganaste! con" intentos " de intentos"
```

FIN



## Codigo

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>


int main(){

int intentos=0,num,aleatorio;

// generar una diferente "semilla" para que cada vez que se ejecute el programa
sea un aleatorio diferente
srand(time(NULL));
aleatorio = rand() % 101;


    // ciclo del juego
    do{

        printf("adivina el numero: ");
        scanf("%d",&num);
        intentos++;


        if (num>aleatorio)
            printf("el numero debe ser menor \n ");
        else if(num<aleatorio)
            printf("el numero debe ser mayor \n ");


    } while ( aleatorio!=num );


    printf("Ganaste!\n con %d intentos\n",intentos);
    return 0;

}
```



## Problema 2

### Enunciado

Desarrollar un programa que gestione las calificaciones de un grupo de cinco estudiantes

El programa debe permitir al usuario ingresar el nombre de cada estudiante y sus calificaciones en tres materias.

Después de ingresar la información de todos los estudiantes, el programa debe calcular el promedio de cada uno de los estudiantes.

Al final el programa debe mostrar una lista con el nombre del estudiante, su promedio y si este aprueba o reprueba según su promedio. El promedio debe ser mayor o igual a 7 para aprobar

### Pseudocódigo

INICIO

Definir N\_ALUMN = 5, N\_CAL = 3

Inicializar arreglos nombres[N\_ALUMN], calif[N\_ALUMN][N\_CAL],  
promedios[N\_ALUMN]

Definir mensaje

For i desde 0 hasta N\_ALUMN - 1

Print "Ingresa el nombre del estudiante:"

Leer nombres[i]

For j desde 0 hasta N\_CAL - 1

Print "Ingresa la calificación j+1:"

Leer calif[i][j]

Sumar calif[i][j] a promedios[i]

Fin For

Calcular promedios[i] dividiendo entre N\_CAL

Fin For



```
Print "Nombre | Promedio | Estado"
For i desde 0 hasta N_ALUMN - 1
    Si promedios[i] >= 7
        mensaje = "Aprobó"
    Sino
        mensaje = "Reprobó"
    Fin Si
    Print nombres[i], promedios[i], mensaje
Fin For
FIN
```

## Código

```
#include <stdio.h>
#define N_ALUMN 5
#define N_CAL 3
int main(){

    char nombres[N_ALUMN][100]; // array de los nombres
    float calif[N_ALUMN][N_CAL]; // filasXcolumnas / alumnosXcalif
    float promedios[5]={0}; // array de los promedios
    char* mensaje; // variable para decir aprobo o reprobó

    for (int i = 0; i < N_ALUMN; i++){
        printf("%d.-ingresa el nombre: ",(i+1));
        scanf("%s",nombres[i]);
        printf("Alumno: %s\n",nombres[i]);
```



```
        for (int j = 0; j < N_CAL; j++){
            printf(" calif %d: ",(j+1) );
            scanf("%f",&calif[i][j] );
            promedios[i]+=calif[i][j];
            // printf("%f\t",calif[i][j] );
        }
        promedios[i]/=N_CAL;
    }

    printf("\nNombre \t Promedio \t Estado\n");
    for (int i = 0; i < N_ALUMN; i++){

        if(promedios[i]>=7)
            mensaje="aprobo";
        else
            mensaje="reprobo";

        printf("%s \t %.3f \t %s \n",nombres[i],promedios[i],mensaje);
    }

    return 0;
}
```



## Problema 3

### Enunciado

Desarrollar un programa que realice tareas con arreglos y cadenas

El programa debe preguntar al usuario si quiere realizar una tarea con arreglos o con cadenas.

Si el usuario elige arreglos, debe preguntarle cuantos elementos tiene el arreglo (máximo 10) y pedirle que ingrese los valores uno a uno. Cuando termine de ingresarlos, el programa debe mostrar en la pantalla el valor más grande del arreglo, el valor más pequeño y el promedio de todos los elementos

Si el usuario elige cadenas, debe pedirle que ingrese la cadena y después mostrar en la pantalla la cantidad de caracteres que tiene y cuantas vocales tiene

### Pseudocódigo

INICIO

    Inicializar opcion = 0, n = 0

Do

    Print "Seleccione una opción 0 = arreglos, 1 = cadenas"

    Leer opcion

Hasta que opcion = 0 o 1

Si opcion = 1 (Cadenas)

    Inicializar cadena[100], vocales = 0, caracteres = 0

    Print "Ingrese la cadena"

    Leer cadena

    For cada carácter en cadena

        Si carácter es vocal (a, e, i, o, u)

            Incrementar vocales

        Fin Si

    Incrementar caracteres

Fin For





Print la cantidad de vocales y caracteres

Sino (Arreglos)

Do

Print "Ingrese el número de elementos (máximo 10)"

Leer n

Hasta que  $1 \leq n \leq 10$

Inicializar vector[n], max, min, prom = 0

Print "Llene el vector"

For i desde 0 hasta n - 1

Print "Ingrese el elemento (i+1)"

Leer vector[i]

Sumar vector[i] a prom

Fin For

Ordenar vector usando método burbuja

Asignar min = primer elemento, max = último elemento

Dividir prom entre n

Print min, max, prom y el vector ordenado

Fin Si

FIN



## Código

```
#include <stdio.h>

#include <ctype.h>

int main(){
    int opcion=0,n=0;

    do{
        printf("Seleccione una opcion:\n 0=trabajar con arreglos \n 1=trabajar con cadenas \nopcion: ");
        scanf("%d",&opcion);
    } while (opcion!=0 && opcion!=1);

    if (opcion){ //opcion=1 (true)

        char cadena[100];
        int vocales=0,caracteres=0;
        printf("\n--- CADENAS ---\n");
        printf("ingrese la cadena: ");
        // fgets(cadena,longitud, stdin);
        getchar();
        fgets(cadena,100, stdin);

        // '\0' es el ultimo valor en una cadena de caracteres
        int i=0;
        while (cadena[i]!='\0'){
            if (cadena[i] == '\n') {
```



```
        break;
    }

    // con tolower convertimos a minuscula el caracter en la posicon
    if (tolower(cadena[i])=='a' || tolower(cadena[i])=='e'
        || tolower(cadena[i])=='i' || tolower(cadena[i])=='o'
        || tolower(cadena[i])=='u' )
        vocales++;

    // ignorar los espacios al hacer el conteo de caracteres
    if( cadena[i]!=' ')
        caracteres++;
    i++;
}

printf("la cadena %s tiene: %d vocales y %d caracteres (sin contar
espacios) \n",cadena,vocales,caracteres);

}else{
    printf("\n--- ARREGLOS ---\n");
    do{
        printf("ingrese el numero de elementos: ");
        scanf("%d",&n);
    } while (n<=0 || n>10);

    float vector[n], aux=0,max,min,prom=0;

    // llenar vector
```



```
printf("Llene el vector: \n");  
for (int i = 0; i < n; i++){  
    printf("[%d]: ",(i+1));  
    scanf("%f",&vector[i]);  
    prom+=vector[i];  
}
```

```
//ordenarlo por el metodo burbuja:  
for (int i = 0; i < n-1; i++) {  
    for (int j = 0; j < n-i-1; j++) {  
        if (vector[j] > vector[j+1]) {  
            aux = vector[j];  
            vector[j] = vector[j+1];  
            vector[j+1] = aux;  
        }  
    }  
}
```

```
// guardar el menor, mayor y dividir el promedio entre el numero de  
elementos
```

```
min=vector[0];  
max=vector[n-1];  
prom/=n;
```

```
printf("\nValor minimo: %.3f",min);  
printf("\nValor maximo: %.3f",max);  
printf("\nPromedio: %.3f",prom);
```



```
        printf("\nVector ordenado:\n");
        for (int i = 0; i < n; i++){
            printf("%.3f \t",vector[i] );
        }

    }

    return 0;
}
```