FACULTAD DE

INGENIERÍA

**INGENIERÍA EN COMPUTACION INGENIERIA DE SISTEMAS**

**INGENIERIA ELECTRONICA**

**2018**





GUÍA DE TRABAJOS PRÁCTICOS

**Alumnos**:

* **MIRANDA, Matias**
* **TUPAC YUPANQUI, VALENTIN, David Elí**

PROGRAMACION II

Prof.: Ing. María Alejandra Bosio



**TRABAJOS PRACTICOS DE AULA**

INDICE

[TRABAJO PRACTICO N.º 1 - Programación en C++ 3](#_Toc515254083)

[1. Objetivos de aprendizaje. 3](#_Toc515254084)

[2. Unidad temática que incluye este trabajo práctico. 3](#_Toc515254085)

[3. Consignas para desarrollar en el trabajo práctico. 3](#_Toc515254086)

[I. Estructuras secuenciales, de bifurcación y repetitivas. 3](#_Toc515254087)

[II. Funciones. 10](#_Toc515254088)

[III. Arreglos. 10](#_Toc515254089)

[IV. Estructuras. 10](#_Toc515254090)

[TRABAJO PRACTICO N.º 2 - Punteros. 10](#_Toc515254091)

[1. Objetivos de aprendizaje. 10](#_Toc515254092)

[2. Unidad temática que incluye este trabajo práctico. 10](#_Toc515254093)

[3. Consignas para desarrollar en el trabajo práctico. 10](#_Toc515254094)

[TRABAJO PRACTICO N.º 3 – Clases y Herencia. 10](#_Toc515254095)

[1. Objetivos de aprendizaje. 11](#_Toc515254096)

[2. Unidad temática que incluye este trabajo práctico. 11](#_Toc515254097)

[3. Consignas para desarrollar en el trabajo práctico. 11](#_Toc515254098)

[TRABAJO PRACTICO N.º 4 – Archivos y Manejo de Excepciones 11](#_Toc515254099)

[1. Objetivos de aprendizaje. 11](#_Toc515254100)

[2. Unidad temática que incluye este trabajo práctico. 11](#_Toc515254101)

[3. Consignas para desarrollar en el trabajo práctico. 11](#_Toc515254102)

[TRABAJO PRACTICO EVALUABLE. 11](#_Toc515254103)

### TRABAJO PRACTICO N.º 1 - Programación en C++

### Objetivos de aprendizaje.

1. Repasar los conceptos de programación ya adquiridos en asignaturas anteriores.
2. Adquirir habilidad en la resolución de problemas y optimización del código.

### Unidad temática que incluye este trabajo práctico.

Este trabajo práctico corresponde a la Unidad 1 de la programación de la asignatura.

### Consignas para desarrollar en el trabajo práctico.

### Estructuras secuenciales, de bifurcación y repetitivas.

|  |
| --- |
| /\* |
|  | **TP1-I-01:** |
|  | Dada una medida de tiempo expresada en horas, minutos y segundos |
|  | con valores arbitrarios, obtenga un programa que transforme |
|  | dicha medida en una expresión correcta. |
|  | Por ejemplo, dada la medida 3h 118m 195s, el programa deberá |
|  | obtener como resultado 5h 1m 15s. |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | int h,m,s; |
|  | cout<<"Ingrese el tiempo "<<endl; |
|  | cout<<"Horas "; cin>>h; |
|  | cout<<"Minutos: "; cin>>m; |
|  | cout<<"Segundos: "; cin>>s; |
|  | s=s+(m\*60)+(h\*3600); |
|  | m=s/60; |
|  | s=s%60; |
|  | h=m/60; |
|  | m=m%60; |
|  | cout<<"El tiempo correctamente expresado:"<<endl; |
|  | cout<<h<<"(hs) "<<m<<"(min) "<<s<<"(seg)"; |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-02:** |
|  | Dado un número verificar: |
|  | - Que tenga dos dígitos |
|  | - Que sus dígitos sean pares |
|  | En caso afirmativo calcular el Promedio de sus dígitos. |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  |  |
|  | int numero; |
|  | int decena = 0; |
|  | int unidad = 0; |
|  | float promedio = 0.0; |
|  |  |
|  | cout<<"Ingrese un valor: "; cin>>numero; |
|  |  |
|  | if((numero / 10) < 10){ |
|  | unidad = (numero%10); |
|  | if(unidad%2 == 0){ |
|  | decena = (numero - unidad)/10; |
|  | if(decena%2 == 0){ |
|  |  |
|  | cout<<"decena: "<<decena<<"\n"; |
|  | cout<<"unidad: "<<unidad<<"\n"; |
|  | promedio = decena + unidad; |
|  | promedio /= 2; |
|  |  |
|  | cout<<"El promedio es: "<<promedio<<"\n"; |
|  |  |
|  | }else{ |
|  | cout<<"La decena no es par\n"; |
|  | } |
|  | }else{ |
|  | cout<<"La unidad no es par\n"; |
|  | } |
|  | }else if(numero > 1){ |
|  | cout<<"El numero ingresado tiene mas de dos digitos\n"; |
|  | }else if(numero == 0){ |
|  | cout<<"El numero ingresado tiene menos de dos digitos\n"; |
|  | } |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-03:** |
|  | Validar el día, mes y el año para una fecha suministrada |
|  | sabiendo que los años van desde 1960 hasta 2021. |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | int dia, mes, ano; |
|  | cout<<"Exprese dia, mes y anio\n"; |
|  | do{ |
|  | cout<<"Dia: ";cin>>dia; |
|  | if(dia < 0 || dia > 31){ |
|  | cout<<"Exprese un dia comprendido entre 1 a 31\n"; |
|  | } |
|  | }while(dia < 0 || dia > 31); |
|  | do{ |
|  | cout<<"Mes: ";cin>>mes; |
|  | if(mes < 0 || mes > 12){ |
|  | cout<<"Exprese un mes comprendido entre 1 a 12\n"; |
|  | } |
|  | }while(mes < 0 || mes > 12); |
|  | do{ |
|  | cout<<"Anio: ";cin>>ano; |
|  | if(ano < 1960 ||ano > 2021){ |
|  | cout<<"Exprese un anio comprendido entre 1960 a 2021\n"; |
|  | } |
|  | }while(ano < 1960 || ano > 2021); |
|  | cout<<"\nUsted ingreso Dia:"<<dia<<" Mes:"<<mes<<" Anio:"<<ano<<"\n"; |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-04:** |
|  | Escriba un programa, que dada una hora del día en Horas, Minutos y Segundos |
|  | calcule la hora correspondiente 15 segundos después. |
|  | (Validar la entrada de datos). |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | int horas, minutos, segundos; |
|  |  |
|  | cout<<"Exprese horas, minutos, segundos\n"; |
|  |  |
|  | do{ |
|  | cout<<"Horas: ";cin>>horas; |
|  | if(horas < 0) cout<<"Ingrese un valor mayor o igual a 0.\n"; |
|  | }while(horas < 0); |
|  |  |
|  | do{ |
|  | cout<<"Minutos: ";cin>>minutos; |
|  | if(minutos < 0) cout<<"Ingrese un valor mayor o igual a 0.\n"; |
|  | }while(minutos < 0); |
|  |  |
|  | do{ |
|  | cout<<"Segundos: ";cin>>segundos; |
|  | if(segundos < 0) cout<<"Ingrese un valor mayor o igual a 0.\n"; |
|  | }while(segundos < 0); |
|  |  |
|  | cout<<"Usted ingreso "<<horas<<"h "<<minutos<<"m "<<segundos<<"s"; |
|  | cout<<" mas 15 segundos despues: \n"; |
|  |  |
|  | segundos += 15; |
|  |  |
|  | while(segundos >= 60){ |
|  | segundos -= 60; |
|  | minutos++; |
|  | } |
|  |  |
|  | while(minutos >= 60){ |
|  | minutos -= 60; |
|  | horas++; |
|  | } |
|  |  |
|  | cout<<"El tiempo final es: "<<horas<<"h "<<minutos<<"m "<<segundos<<"s.\n"; |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-05:** |
|  | La fecha de Pascua corresponde al primer domingo después de la primera |
|  | luna llena que sigue al equinoccio de primavera, y se calcula con |
|  | las siguientes expresiones: |
|  | A = año mod 19 |
|  | B = año mod 4 |
|  | C = año mod 7 |
|  | D = (19\*A + 24) mod 30 |
|  | E = (2\*B + 4\*C + 6\*D + 5) mod 7 |
|  | N = (22 + D + E) |
|  | En el que N indica el número del día de marzo (o Abril si N > 31) |
|  | correspondiente al domingo de Pascua. Realizar un programa que |
|  | dado un año calcule esa fecha. |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | int ano; |
|  | int A, B, C, D, E, N; |
|  |  |
|  | cout<<"Exprese el anio\n"; |
|  |  |
|  | do{ |
|  | cin>>ano; |
|  | if(ano < 0 ||ano > 9999){ |
|  | cout<<"Exprese un anio comprendido entre 0 a 9999\n"; |
|  | } |
|  | }while(ano < 0 || ano > 9999); |
|  |  |
|  |  |
|  | A = ano % 19; |
|  | B = ano % 4; |
|  | C = ano % 7; |
|  | D = (19\*A + 24) % 30; |
|  | E = (2\*B + 4\*C + 6\*D + 5) % 7; |
|  | N = (22 + D + E); |
|  |  |
|  | if(N > 31){ |
|  | cout<<"La fecha de pascuas es: "<<(N-31)<<" de abril del "<<ano<<"\n"; |
|  | }else{ |
|  | cout<<"La fecha de pascuas es: "<<N<<" de marzo del "<<ano<<"\n"; |
|  | } |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-06:** |
|  | Calcular el valor de PI aplicando la fórmula: |
|  | PI = 4\*(1 - 1/3 + 1/5 - 1/7 + ... - 1/99) |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  | #include <math.h> |
|  | #include <iomanip> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | double pi=0; |
|  |  |
|  | for(int n=0;n<=99;n++){ |
|  | pi = pi + 4\*(pow(-1,n)/(2\*n+1)); |
|  | } |
|  | cout<<"PI = 4\*(1 - 1/3 + 1/5 - 1/7 + ... - 1/99)"<<endl; |
|  | cout<<setprecision(10)<<fixed; |
|  | cout<<"El valor de PI es: "<<pi<<endl; |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-07:** |
|  | Obtener un programa que calcule el factorial de un número n |
|  | \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<conio.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main(){ |
|  | int numero; |
|  | unsigned long long factorial=1; |
|  | cout<<"\nIngrese un numero: ";cin>>numero; |
|  | for(int i = 1; i <= numero; i++){ |
|  | factorial = factorial \* i; |
|  | } |
|  | cout<<"\nEl factorial de: "<<numero<<"! = "<<factorial<<endl; |
|  | getch(); |
|  | return 0; |
|  | } |

|  |
| --- |
| /\* |
|  | **TP1-I-08:** |
|  | Obtener un programa que calcule y muestre e^x |
|  | utilizando la serie: |
|  | e^x = 1 + x/1! + x^2/2! + x^3/3! + ... + x^n/n! |
|  | \*/ |
|  |  |
|  | #include <iostream> |
|  | #include <conio.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main() |
|  | { |
|  | int contador=1; |
|  | int nTerminos; |
|  |  |
|  | double total=1; |
|  | double s=1; |
|  | double m=1; |
|  | double x; |
|  |  |
|  | cout <<"Ingres el valor de x: "; |
|  | cin >> x; |
|  | cout <<"Ingrese los 'n' terminos de la serie: "; |
|  | cin >> nTerminos; |
|  |  |
|  | while (contador <= nTerminos) |
|  | { |
|  | s \*= x; |
|  | m \*= contador; |
|  | total += (s/m); |
|  | contador++; |
|  | } |
|  | cout<<"e^x = 1 + x/1! + x^2/2! + x^3/3! + ... + x^n/n!"<<endl; |
|  | cout << "e^"<<x<<" = " << total; |
|  |  |
|  | getch(); |
|  | return 0; |
|  | } |

### Funciones.

### Arreglos.

### Estructuras.

### TRABAJO PRACTICO N.º 2 - Punteros.

### Objetivos de aprendizaje.

1. Comprender el significado de los punteros y su relación con arreglos y cadenas.
2. Desarrollar programas utilizando gestión dinámica de memoria.

### Unidad temática que incluye este trabajo práctico.

Este trabajo práctico corresponde a la unidad 3 de la programación de la asignatura.

Los ejercicios 3, 5, 6, 7, 9, y 11 son de presentación obligatoria.

Los demás son de presentación optativa.

### Consignas para desarrollar en el trabajo práctico.

### TRABAJO PRACTICO N.º 3 – Clases y Herencia.

### Objetivos de aprendizaje.

1. Comprender el concepto de Orientación a Objetos y tipos abstractos.
2. Comprender el concepto de herencia.
3. Construir programas en C++ utilizando definiciones de clases y herencia.

### Unidad temática que incluye este trabajo práctico.

Este trabajo práctico corresponde a las unidades 4,5,6 y 7 de la programación de la asignatura.

Los ejercicios 3, 4, 7, 8, 9 y 10 son de presentación obligatoria. Los demás son de presentación optativa.

### Consignas para desarrollar en el trabajo práctico.

### TRABAJO PRACTICO N.º 4 – Archivos y Manejo de Excepciones

### Objetivos de aprendizaje.

1. Comprender el manejo de archivos y excepciones.
2. Construir programas en C++ utilizando archivos.
3. Construir programas con manejo de excepciones.

### Unidad temática que incluye este trabajo práctico.

Este trabajo práctico corresponde a la unidad 8 de la programación de la asignatura.

Los ejercicios 2 y 3 son de presentación obligatoria. Los demás son de presentación optativa.

### Consignas para desarrollar en el trabajo práctico.

### TRABAJO PRACTICO EVALUABLE.