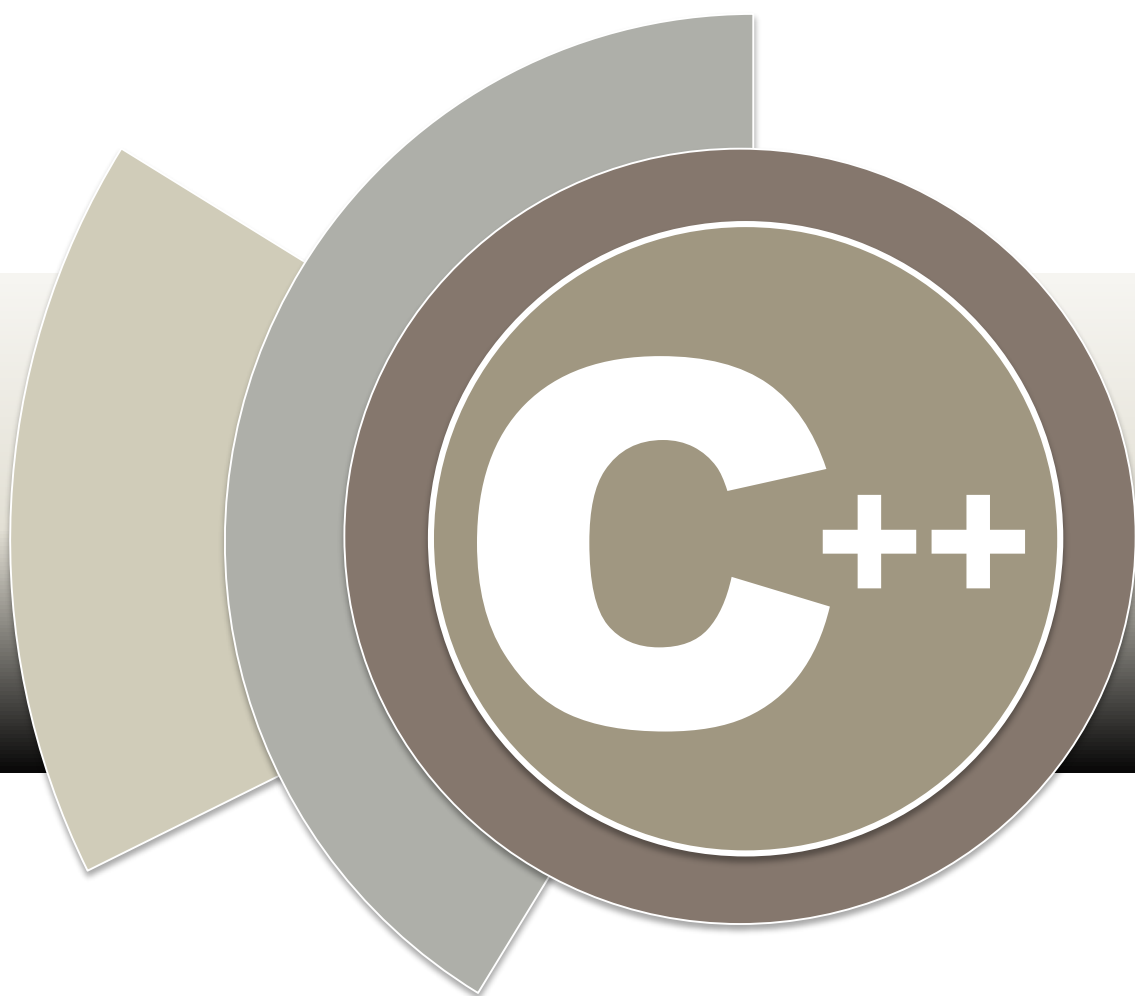




Programación en



Libro práctico de Programación en C++



Bloque 1: Introducción y Entrada-Salida

Ejercicio 1: Escribe un programa que lea de la entrada estándar dos números y muestre en la salida estándar su suma, resta, multiplicación y división.

Ejercicio 2: Escribir un programa que de la entrada estándar el precio de un producto y muestre en la salida estándar el precio del producto al aplicarle el IVA.

Ejercicio 3: Realice un programa que lea de la entrada estándar los siguientes datos de una persona:

Edad: dato de tipo entero.

Sexo: dato de tipo carácter.

Altura en metros: dato de tipo real.

Tras leer los datos, el programa debe mostrarlos en la salida estándar.

Ejercicio 4: Ejecute el programa del ejercicio anterior con entradas erróneas y observe los resultados. Por ejemplo, introduzca un dato de tipo carácter cuando se espera un dato de tipo entero.

Bloque 2: Expresiones y Operadores

Ejercicio 1: Escribe la siguiente expresión como expresión en C++:

$$\mathbf{a)} \quad \frac{a}{b} + 1$$

Ejercicio 2: Escribe la siguiente expresión como expresión en C++:

$$\mathbf{b)} \quad \frac{a+b}{c+d}$$

Ejercicio 3: Escribe la siguiente expresión como expresión en C++:

$$\mathbf{c)} \quad \frac{a + \frac{b}{c}}{d + \frac{e}{f}}$$

Ejercicio 4: Escribe la siguiente expresión como expresión en C++:

$$\mathbf{d)} \quad a + \frac{b}{c-d}$$

Ejercicio 5: Escriba un fragmento de programa que intercambie los valores de dos variables.



Ejercicio 6: Escriba un programa que lea las tres notas de un alumno y calcule la nota final media de dicho alumno.

Ejercicio 7: La calificación final de un estudiante es el promedio de tres notas: la nota de prácticas que cuenta un 30% del total, la nota teórica que cuenta un 60% y la nota de participación que cuenta el 10% restante. Escriba un programa que lea las tres notas del alumno y escriba su nota final.

Ejercicio 8: Escriba un programa que lea de la entrada estándar los dos catetos de un triángulo rectángulo y escriba en la salida estándar su hipotenusa.

Ejercicio 9: Realice un programa que calcule el valor que toma la siguiente función para unos valores dados de x e y :

$$f(x, y) = \frac{\sqrt{x}}{y^2 - 1}$$

Ejercicio 10: Escriba un programa que calcule las soluciones de una ecuación de segundo grado de la forma $ax^2 + bx + c = 0$, teniendo en cuenta que:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Bloque 3: Estructuras Condicionales

Ejercicio 1: Escriba un programa que lea dos números y determine cuál de ellos es el mayor.

Ejercicio 2: Escriba un programa que lea tres números y determine cuál de ellos es el mayor.

Ejercicio 3: Realice un programa que lea un valor entero y determine si se trata de un número par o impar.

Ejercicio 4: Comprobar si un número digitado por el usuario es positivo o negativo.

Ejercicio 5: Escriba un programa que lea de la entrada estándar un carácter e indique en la salida estándar si el carácter es una vocal minúscula o no.

Ejercicio 6: Escriba un programa que lea de la entrada estándar un carácter e indique en la salida estándar si el carácter es una vocal minúscula, es una vocal mayúscula o no es una vocal.

Ejercicio 7: Escriba un programa que solicite una edad (un entero) e indique en la salida estándar si la edad introducida está en el rango [18-25].

Ejercicio 8: Escriba un programa que lea de la entrada estándar tres números. Después debe leer un cuarto número e indicar si el número coincide con alguno de los introducidos con anterioridad.



Ejercicio 9: Cambiar un número entero con el mismo valor pero en romanos.

Ejercicio 10: Mostrar los meses del año, pidiéndole al usuario un número entre (1-12), y mostrar el mes al que corresponde.

Ejercicio 11: Hacer un programa que simule un cajero automático con un saldo inicial de 1000 Dólares.

Ejercicio 12: Hacer un menú que considere las siguientes opciones:

Caso 1: Cubo de un número

Caso 2: Número par o impar

Caso 3: salir.

Bloque 4: Estructuras Repetitivas

Ejercicio 1: Realice un programa que solicite de la entrada estándar un entero del 1 al 10 y muestre en la salida estándar su tabla de multiplicar.

Ejercicio 2: Realice un programa que lea de la entrada estándar números hasta que se introduzca un cero. En ese momento el programa debe terminar y mostrar en la salida estándar el número de valores mayores que cero leídos.

Ejercicio 3: Realice un programa que calcule y muestre en la salida estándar la suma de los cuadrados de los 10 primeros enteros mayores que cero.

Ejercicio 4: Escriba un programa que tome cada 4 horas la temperatura exterior, leyéndola durante un período de 24 horas. Es decir, debe leer 6 temperaturas. Calcule la temperatura media del día, la temperatura más alta y la más baja.

Ejercicio 5: Escriba un programa que lea valores enteros hasta que se introduzca un valor en el rango [20-30] o se introduzca el valor 0. El programa debe entregar la suma de los valores mayores a 0 introducidos.

Ejercicio 6: Escriba un programa que calcule x^y , donde tanto x como y son enteros positivos, sin utilizar la función `pow()`.

Ejercicio 7: Escriba un programa que calcule el valor de: $1+2+3+\dots+n$

Ejercicio 8: Escriba un programa que calcule el valor de: $1+3+5+\dots+2n-1$

Ejercicio 9: Escriba un programa que calcule el valor de: $1*2*3*\dots*n$ (factorial)

Ejercicio 10: Escriba un programa que calcule el valor de: $1!+2!+3!+\dots+n!$ (suma de factoriales).

Ejercicio 11: Escriba un programa que calcule el valor de: $2^1+2^2+2^3+\dots+2^n$

Ejercicio 12: Hacer un programa que calcule el resultado de la siguiente expresión:
 $1-2+3-4+5-6\dots n$

Ejercicio 13: Hacer un que realice la serie fibonacci -> 1 1 2 3 5 8 13...n



Ejercicio 14: En una clase de 5 alumnos se han realizado tres exámenes y se requiere determinar el número de:

- a) Alumnos que aprobaron todos los exámenes.
- b) Alumnos que aprobaron al menos un examen.
- c) Alumnos que aprobaron únicamente el último examen.

Realice un programa que permita la lectura de los datos y el cálculo de las estadísticas.

Ejercicio 15: Realice un programa que solicite al usuario que piense un número entero entre el 1 y el 100. El programa debe generar un número aleatorio en ese mismo rango [1-100], e indicarle al usuario si el número que digito es menor o mayor al número aleatorio, así hasta que lo adivine. Y por último mostrarle el número de intentos que le llevo.

Ejercicio 16: Realice un programa que calcule la descomposición en factores primos de un número entero. Por ejemplo: $20 = 2 \cdot 2 \cdot 5$.

Bloque 5: Arreglos o Vectores

Ejercicio 1: Escribe un programa que defina un vector de números y calcule la suma de sus elementos.

Ejercicio 2: Escribe un programa que defina un vector de números y calcule la multiplicación acumulada de sus elementos.

Ejercicio 3: Escribe un programa que lea de la entrada estándar un vector de números y muestre en la salida estándar los números del vector con sus índices asociados.

Ejercicio 4: Escribe un programa que defina un vector de números y muestre en la salida estándar el vector en orden inverso—del último al primer elemento.

Ejercicio 5: Desarrolle un programa que lea de la entrada estándar un vector de enteros y determine el mayor elemento del vector.

Ejercicio 6: Escribe un programa que defina un vector de números y calcule si existe algún número en el vector cuyo valor equivale a la suma del resto de números del vector.

Ejercicio 7: Realiza un programa que defina dos vectores de caracteres y después almacene el contenido de ambos vectores en un nuevo vector, situando en primer lugar los elementos del primer vector seguido por los elementos del segundo vector. Muestre el contenido del nuevo vector en la salida estándar.

Ejercicio 8: Hacer un programa que lea 5 números en un arreglo, los copie a otro arreglo, multiplicado por 2 y muestre el segundo arreglo.

Bloque 6: Matrices o Tablas

Ejercicio 1: Hacer un programa para rellenar una matriz pidiendo al usuario el número de filas y columnas, Posteriormente mostrar la matriz en pantalla.



Ejercicio 2: Realiza un programa que defina una matriz de 3x3 y escriba un ciclo para que muestre la diagonal principal de la matriz.

Ejercicio 3: Hacer una matriz de tipo entera de 2 * 2, llenarla de números y luego copiar todo su contenido hacia otra matriz.

Ejercicio 4: Hacer una matriz preguntando al usuario el número de filas y columnas, llenarla de números aleatorios, copiar el contenido a otra matriz y por último mostrarla en pantalla.

Ejercicio 5: Realice un programa que lea una matriz de 3x3 y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

Ejercicio 6: Realice un programa que calcule la suma de dos matrices cuadradas de 3x3.

Ejercicio 7: Desarrollar un programa que determine si una matriz es simétrica o no. Una matriz es simétrica si es cuadrada y si es igual a su matriz traspuesta.

Ejercicio 8: Realice un programa que calcule el producto de dos matrices cuadradas de 3x3.

Bloque 7: Cadenas

Ejercicio 1: Hacer un programa que pida al usuario que digite una cadena de caracteres, luego verificar la longitud de la cadena, y si ésta supera a 10 caracteres mostrarla en pantalla, caso contrario no mostrarla.

Ejercicio 2: Pedir al usuario una cadena de caracteres, almacenarla en un arreglo y copiar todo su contenido hacia otro arreglo de caracteres.

Ejercicio 3: Pedir al usuario que digite 2 cadenas de caracteres, e indicar si ambas cadenas son iguales, en caso de no serlo, indicar cuál es mayor alfabéticamente.

Ejercicio 4: Crear una cadena que tenga la siguiente frase "Hola que tal", luego crear otra cadena para preguntarle al usuario su nombre, por ultimo añadir el nombre al final de la primera cadena y mostrar el mensaje completo "Hola que tal (NombreDelUsuario)".

Ejercicio 5: Hacer un programa que determine si una palabra es palíndroma.

Ejercicio 6: Convertir dos cadenas de minúsculas a MAYÚSCULAS. Compararlas, y decir si son iguales o no.

Ejercicio 7: Pedir su nombre al usuario en MAYÚSCULA, si su nombre comienza por la letra A, convertir su nombre a minúsculas, caso contrario no convertirlo.

Ejercicio 8: Pedir al usuario 2 cadenas de caracteres de números, uno entero y el otro real, convertirlos a sus respectivos valores y por ultimo sumarlos.



Ejercicio 9: Realice un programa que lea una cadena de caracteres de la entrada estándar y muestre en la salida estándar cuántas ocurrencias de cada vocal existen en la cadena.

Bloque 8: Estructuras

Ejercicio 1: Hacer una estructura llamada corredor, en la cual se tendrán los siguientes campos: Nombre, edad, sexo, club, pedir datos al usuario para un corredor, y asignarle una categoría de competición:

- Juvenil ≤ 18 años
- Señor ≤ 40 años
- Veterano > 40 años

Posteriormente imprimir todos los datos del corredor, incluida su categoría de competición.

Ejercicio 2: Hacer una estructura llamada alumno, en la cual se tendrán los siguientes Campos: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.

Ejercicio 3: Realizar un programa que lea un arreglo de estructuras los datos de N empleados de la empresa y que imprima los datos del empleado con mayor y menor salario.

Ejercicio 4: Hacer un arreglo de estructura llamada atleta para N atletas que contenga los siguientes campos: nombre, país, numero de medallas. Y devuelva los datos (Nombre, país) del atleta que ha ganado el mayor número de medallas.

Ejercicio 5: Hacer 2 estructuras una llamada promedio que tendrá los siguientes campos: nota1, nota2, nota3; y otro llamada alumno que tendrá los siguientes miembros: nombre, sexo, edad; hacer que la estructura promedio este anidada en la estructura alumno, luego pedir todos los datos para un alumno, luego calcular su promedio, y por ultimo imprimir todos sus datos incluidos el promedio.

Ejercicio 6: Utilizar las 2 estructuras del problema 5, pero ahora pedir los datos para N alumnos, y calcular cuál de todos tiene el mejor promedio, e imprimir sus datos.

Ejercicio 7: Defina una estructura que indique el tiempo empleado por un ciclista en una etapa. La estructura debe tener tres campos: horas, minutos y segundos. Escriba un programa que dado n etapas calcule el tiempo total empleado en correr todas las etapas.

Ejercicio 8: Defina una estructura que sirva para representar a una persona. La estructura debe contener dos campos: el nombre de la persona y un valor de tipo lógico que indica si la persona tiene algún tipo de discapacidad. Realice un programa que dado un vector de personas rellene dos nuevos vectores: uno que contenga las personas que no tienen ninguna discapacidad y otro que contenga las personas con discapacidad.



Bloque 9: Funciones

Ejercicio 1: Escriba una función llamada `mult()` que acepte dos números en punto flotante como parámetros, multiplique estos dos números y despliegue el resultado.

Ejercicio 2: Escriba una función llamada `al_cuadrado()` que calcule el cuadrado del valor que se le transmite y despliegue el resultado. La función deberá ser capaz de elevar al cuadrado números flotantes.

Ejercicio 3: Escriba una función nombrada `funpot()` que eleve un número entero que se le transmita a una potencia en número entero positivo y despliegue el resultado. El número entero positivo deberá ser el segundo valor transmitido a la función.

Ejercicio 4: Escriba un programa en C++ que devuelva la parte fraccionaria de cualquier número introducido por el usuario. Por ejemplo, si se introduce el número 256.879, debería desplegarse el número 0.879.

Ejercicio 5: Escriba una plantilla de función llamada `despliegue()` que despliegue el valor del argumento único que se le transmite cuando es invocada la función.

Ejercicio 6: Escriba una plantilla de función llamada `máximo()` que devuelva el valor máximo de tres argumentos que se transmitan a la función cuando sea llamada. Suponga que los tres argumentos serán del mismo tipo de datos.

Ejercicio 7: Intercambiar el valor de 2 variables utilizando paso de parámetros por referencia.

Ejercicio 8: Escriba una función nombrada `cambio()` que tenga un parámetro en número entero y seis parámetros de referencia en número entero nombrados cien, cincuenta, veinte, diez, cinco y uno, respectivamente. La función tiene que considerar el valor entero transmitido como una cantidad en dólares y convertir el valor en el número menor de billetes equivalentes.

Ejercicio 9: Escriba una función nombrada `tiempo()` que tenga un parámetro en número entero llamado `totalSeg` y tres parámetros de referencia enteros nombrados horas, min y seg. La función es convertir el número de segundos transmitido en un número equivalente de horas, minutos y segundos.

Ejercicio 10: Escriba una función nombrada `calc_años()` que tenga un parámetro entero que represente el número total de días desde la fecha 1/1/2000 y parámetros de referencia nombrados año, mes y día. La función es calcular el año, mes y día actual para el número dado de días que se le transmitan. Usando las referencias, la función deberá alterar en forma directa los argumentos respectivos en la función que llama. Para este problema suponga que cada año tiene 365 días y cada mes tiene 30 días.

Ejercicio 11: Realice una función que tome como parámetros un vector de números enteros y devuelva la suma de sus elementos.

Ejercicio 12: Realice una función que tome como parámetros un vector y su tamaño y diga si el vector está ordenado crecientemente. Sugerencia: compruebe que para todas las posiciones del vector, salvo para la 0, el elemento del vector es mayor o igual que el elemento que le precede en el vector.



Ejercicio 13: Realiza una función que tome como parámetros un vector de números y su tamaño y cambie el signo de los elementos del vector.

Ejercicio 14: Realiza una función que tome como parámetros un vector de enteros y su tamaño e imprima un vector con los elementos impares del vector recibido.

Ejercicio 15: Desarrollar una función que determine si una matriz es simétrica o no.

Ejercicio 16: Realice una función que dada una matriz y un número de fila de la matriz devuelva el menor de los elementos almacenados en dicha fila.

Ejercicio 17: Suma de números Complejos.

Ejercicio 18: Escriba una función en C++ llamada mayor() que devuelva la fecha más reciente de cualquier par de fechas que se le transmitan. Por ejemplo, si se transmiten las fechas 10/9/2005 y 11/3/2015 a mayor(), será devuelta la segunda fecha.

Ejercicio 19: Realice una función recursiva que sume los primeros n enteros positivos. Nota: para plantear la función recursiva tenga en cuenta que la suma puede expresarse mediante la siguiente recurrencia:

$$\text{suma}(n) = 1, \text{ si } n=1 \\ n + \text{suma}(n-1), \text{ si } n > 1$$

Ejercicio 20: Realice una función recursiva para la serie Fibonacci Nota: La serie de Fibonacci está formada por la secuencia de números: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...

Ejercicio 21: Escriba una función recursiva que calcule un número elevado a una potencia entera mayor o igual que cero: x^y .

Ejercicio 22: Escriba una función escribeNumeros(int ini, int fin) que devuelva en la salida estándar los números del ini al fin. Escriba una versión que escriba los números en orden ascendente.

Bloque 10: Punteros

Ejercicio 1: Comprobar si un número es par o impar, y señalar la posición de memoria donde se está guardando el número. Con punteros.

Ejercicio 2: Determinar si un número es primo o no; con puteros y además indicar en qué posición de memoria se guardó el número.

Ejercicio 3: Rellenar un array de 10 números, posteriormente utilizando punteros indicar cuales son números pares y su posición en memoria.

Ejercicio 4: Rellenar un arreglo con n números, posteriormente utilizando punteros determinar el menor elemento del vector.

Ejercicio 5: Pedir al usuario N números, almacenarlos en un arreglo dinámico posteriormente ordenar los números en orden ascendente y mostrarlos en pantalla. NOTA: Utilizar cualquier método de ordenamiento.



Ejercicio 6: Hacer una función para almacenar N números en un arreglo dinámico, posteriormente en otra función buscar un número en particular.

NOTA: Puedes utilizar cualquier método de búsqueda (Secuencial o Binaria)

Ejercicio 7: Pedir su nombre al usuario y devolver el número de vocales que hay.

NOTA: Recuerda que debes utilizar punteros.

Ejercicio 8: Pedir una cadena de caracteres (string) al usuario, e indicar cuantas veces aparece la vocal a,e,i,o,u; en la cadena de caracteres.

NOTA: Usar punteros

Ejercicio 9: Realice un programa que calcule la suma de dos matrices dinámicas.

Ejercicio 10: Realice un programa que lea una matriz dinámica de NxM y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

Ejercicio 11: Hacer una estructura llamada alumno, en la cual se tendrán los siguientes Campos: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.

NOTA: Usar punteros a estructura.

Ejercicio 12: Defina una estructura que indique el tiempo empleado por un ciclista en recorrer una etapa. La estructura debe tener tres campos: horas, minutos y segundos. Escriba un programa que dado 3 etapas calcule el tiempo total empleado en correr todas las etapas. NOTA: Usar punteros.

Bloque 11: Pilas

Ejercicio 1: Hacer un programa para agregar números enteros a una pila, hasta que el usuario lo decida, después mostrar todos los números introducidos en la pila.

Ejercicio 2: Hacer un programa en C++, utilizando Pilas que contenga el siguiente menú:

1. Insertar un carácter a la pila
2. Mostrar todos los elementos de la pila
3. Salir

Bloque 12: Colas

Ejercicio 1: Hacer un programa en C++, utilizando Colas que contenga el siguiente menú:

1. Insertar un carácter a una cola
2. Mostrar todos los elementos de la cola
3. Salir

Ejercicio 2: Hacer un programa que guarde datos de clientes de un banco, los almacene en cola, y por ultimo muestre los clientes en el orden correcto.



Bloque 13: Listas

Ejercicio 1: Crear una lista que almacene "n" números enteros y calcular el menor y mayor de ellos.

Ejercicio 2: Crear una lista que almacene "n" números reales y calcular su suma y promedio.

Bloque 14: Archivos

Ejercicio 1: Realice un programa que pida al usuario el nombre de un fichero de texto y, a continuación permita almacenar al usuario tantas frases como el usuario desee.

Ejercicio 2: Realice un programa que pida al usuario el nombre o ubicación de un fichero de texto y, a continuación de lectura a todo el fichero.

Ejercicio 3: Realice un programa que pida al usuario el nombre o ubicación de un fichero de texto y, a continuación añada texto en el hasta que el usuario lo decida.

Ejercicio 4: Hacer un programa en C++. Para guardar números telefónicos que muestre un menú con las siguientes opciones:

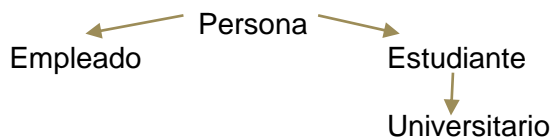
1. Crear (nombre, apellidos, teléfono)
2. Agregar más contactos (nombre, apellidos, teléfono)
3. Visualizar contactos existentes

Bloque 15: Programación Orientada a Objetos (POO)

Ejercicio 1: Construya una clase llamada Rectángulo que tenga los siguientes atributos: largo y ancho, y los siguientes métodos: perímetro() y área()

Ejercicio 2: Construya una clase Tiempo que contenga los siguientes atributos enteros: horas, minutos y segundos. Haga que la clase contenga 2 constructores, el primero debe tener 3 parámetros Tiempo(int,int,int) y el segundo sólo tendrá un campo que serán los segundos y desensamble el número entero largo en horas, minutos y segundos.

Ejercicio 3: Realice un programa en C++, de tal manera que se construya una solución para la jerarquía(herencia) de clases mostrada en la siguiente figura:



Ejercicio 4: Crear un programa en C++ que tenga la siguiente jerarquía de clases: Animal(Clase Padre) -> Humano(Clase Hija) -> Perro(Clase Hija), y hacer polimorfismo con el método comer().