# LENGUAJE DE PROGRAMACIÓN C

PROGRAMACIÓN MODULAR EN C/C++

Ing. Elba Ma. Bodero Poveda, PhD

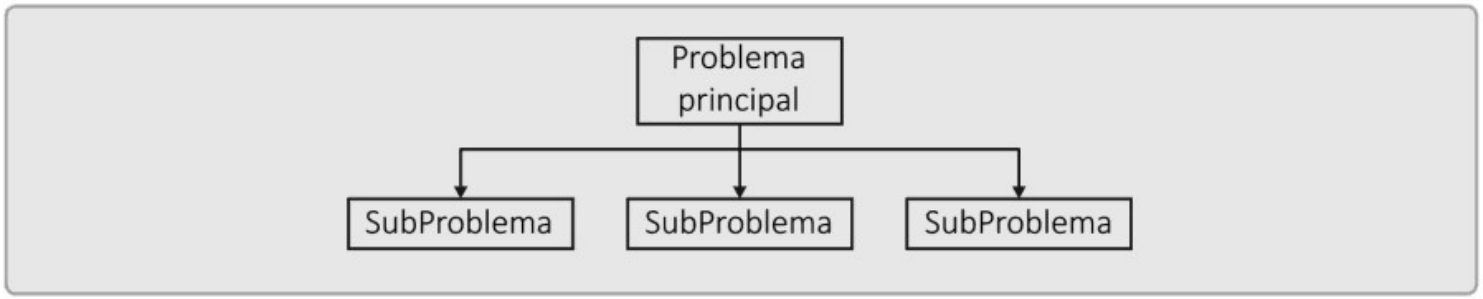
# LENGUAJE DE PROGRAMACIÓN C

SUBALGORITMOS (PROCEDIMIENTOS Y FUNCIONES) EN C/C++

Ing. Elba Ma. Bodero Poveda, PhD

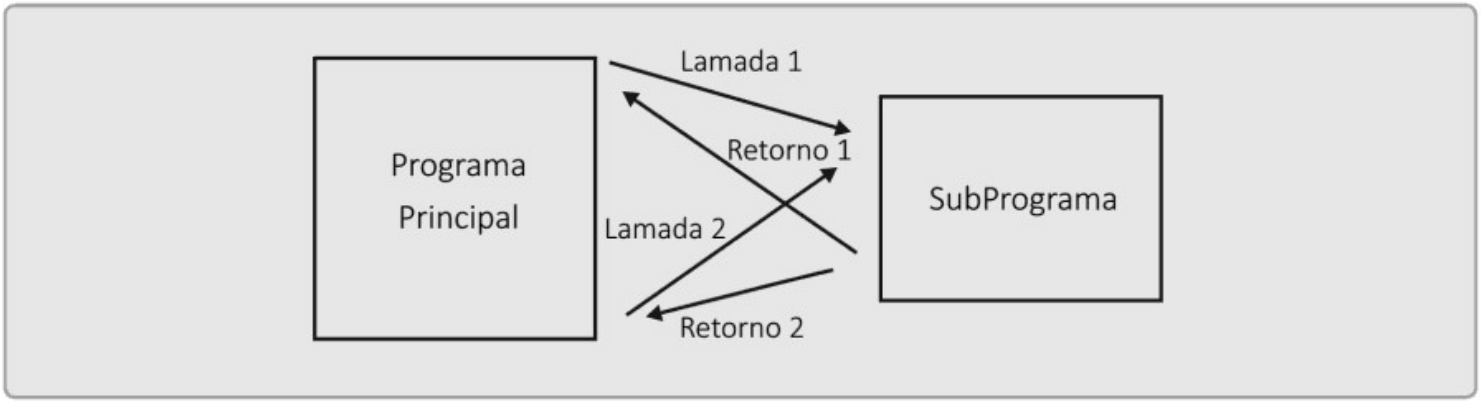
# PROCEDIMIENTOS Y FUNCIONES

El método para diseñar la solución de un problema principal (***main***) en subproblemas se conoce como diseño descendente (***top-dow design***), difundida por la programación modular.



# PROCEDIMIENTOS Y FUNCIONES

El subprograma recibe datos y es invocado desde el programa principal, después de terminar el proceso que tuvo que realizar el subprograma devuelve el resultado correspondiente al programa principal.



# PROCEDIMIENTOS

Los procedimientos se caracterizan por realizar una tarea específica y no retornar un resultado; sin embargo, sí es posible implementar que devuelva resultados por intermedio de parámetros llamados de salida o por referencia.

```
//Función que no retorna ningun valor (void)
void Proc1(int Param1) {

   <Instrucciones>;

}

'Invocar al método
Proc1(10);
```

# FUNCIONES

Son más conocidos por devolver un valor como resultado de la tarea realizada; los lenguajes de programación incorporan funciones que realizan algunas tareas ya programadas, conocidas como funciones internas, pero las funciones programadas por el usuario (programador) se conocen como externas o funciones definidas por el usuario (FDU).

```
//Crear una método que retorna un valor
string Func1(int Param1) {

   <Instrucciones>;

   return <Valor>;

}

//Invocar el método
c = Func1(10);
```

# PARÁMETROS

Muchas veces los procedimientos y funciones requieren que le envíen una lista de valores llamados parámetros (argumentos), para usarlos en la solución de la tarea encomendada. Los parámetros son variables, muchas veces de entrada (reciben valores) y de salida (devuelven resultados) o ambos de entrada/salida.

Estos parámetros también toman el nombre de **parámetros por valor** (entrada) y **parámetros por referencias** (salida).

```
//Crear una método que retorna un valor
string Func1(int Param1) {

   <Instrucciones>;

   return <Valor>;

}
```

# PARÁMETROS POR VALOR (entrada)

Los valores que se envían a los parámetros son asignados como una copia de los valores originales, desconectando el programa principal con el subprograma; es decir, si los valores de los parámetros cambian dentro del subprograma no afecta al programa principal.

```
//Crear un método
int Incrementar(int N) {

   N = N + 1; //Modifica el valor de N

   return  N;

}

//Invocar el método
Num = 5;
Res = Incrementar(Num); //El valor de Num se copia en N
cout<<Num;  //su valor sigue siendo 5
cout<<Res;  //su valor es 6
```

# PARÁMETROS POR REFERENCIA (salida)

Se asignan las referencias de las variables (dirección de memoria de la variable) a los parámetros, conectando el programa principal con el subprograma; es decir, si los valores de los parámetros cambian dentro del subprograma, afecta a las variables del programa principal.

```
//Crear una función
int Incrementar(int &N) {

   N = N + 1; //Modifica el valor de N

   return N;

}
```