UPN

UNIVERSIDAD
PRIVADA
DEL NORTE

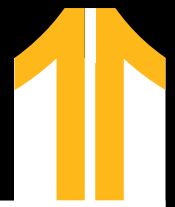
FUNCIONES DEFINIDAS POR EL USUARIO

SEMANA 9

AGENDA

II	**Estructuras de control básicas** Al finalizar la unidad, el estudiante implementa algoritmos para resolver problemas utilizando estructuras de control básicas, como: secuenciales, condicionales y repetitivas—en el lenguaje C#; documentando su trabajo mediante diagramas de flujo o pseudocódigo, y empleando un sistema de control de versiones para gestionar su código y facilitar el desarrollo colaborativo.	**4**	Estructuras Condicionales: simple y doble (D. FLUJO o pseudocódigo y código) Estructura de selección múltiple (D. FLUJO o pseudocódigo y código)
		5	Estructura condicional anidada (D. FLUJO o pseudocódigo y código) Estructura Repetitiva Para (D. FLUJO o pseudocódigo y código)
		6	**Evaluación T1** Estructura Repetitiva: Mientras (D. FLUJO o pseudocódigo y código)
		7	Estructura Repetitiva: Hacer Mientras (D. FLUJO o pseudocódigo y código) Taller de resolución de ejercicios (D. FLUJO o pseudocódigo y código)
		8	Funciones definidas por el usuario. Tipos de funciones (Return, void), Parámetros Alcance de variables: Globales, locales, estáticas
		9	Funciones: Parámetros por valor y por referencia (Métodos) (D. FLUJO o pseudocódigo y código) Funciones de encabezados propias (Bibliotecas)
		10	Taller de desarrollo de casos **Evaluación T2**

LOGRO DE LA SESIÓN



Al finalizar la sesión, el alumno realiza programas aplicando funciones con y sin retorno, así como, variables locales y globales en la solución de problemas.

FUNCIÓN EN PROGRAMACIÓN

En programación, una **función** es una sección de un programa que calcula un valor de manera independiente al resto del programa.
Una función tiene tres componentes importantes:

- Los **parámetros**, que son los valores que recibe la función como entrada.
- El **código de la función**, que son las operaciones que hace la función.
- El **resultado** (o **valor de retorno**), que es el valor final que entrega la función.

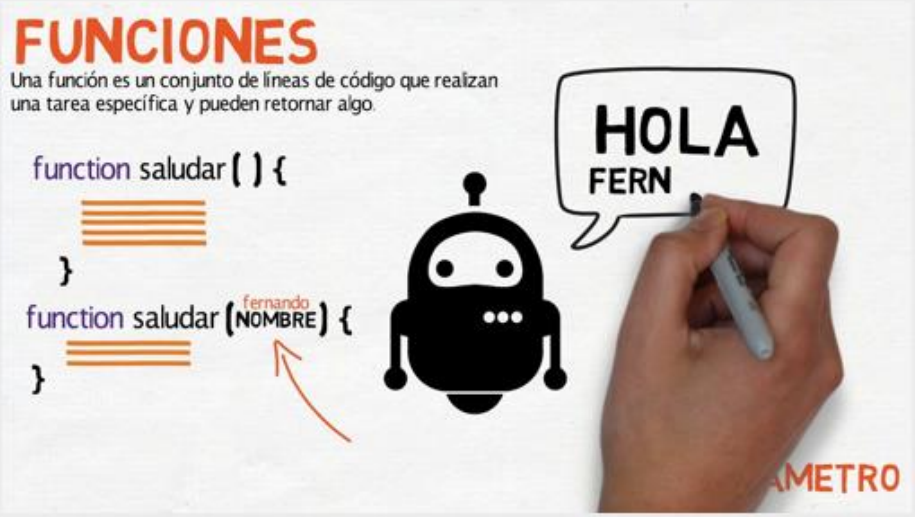

```
function hola(){
    alert("Hola");
}

hola();
```

FUNCIÓN EN PROGRAMACIÓN

Un método con valor de retorno es un módulo de programa que puede recibir datos de entrada a través de variables locales denominadas parámetros y que retorna un resultado al punto donde es invocado. Este tipo de método se utiliza para efectuar cualquier tipo de proceso que produzca un resultado

Estos métodos pueden dividirse a su vez en dos tipos:

- Métodos con valor de retorno sin parámetros
- Métodos con valor de retorno con parámetros



FUNCIÓN EN PROGRAMACIÓN

Este tipo de método no recibe datos de entrada a través de parámetros; pero retorna un valor al punto donde es invocado.

Figura 1 Método con valor de retorno sin parámetros

Este tipo de método se define de la siguiente manera:

```
tiporetorno nombremetodo ( ) {
        Declaración de variables locales
        Cuerpo del método
        return valor;
}
```

Donde:

nombremetodo	: Es el nombre del método.
tiporetorno	: Es el tipo del valor de retorno.
valor	: Es el valor de retorno.

Llamada

Este tipo de método se invoca de la siguiente manera:

```
variable = nombremetodo ( );
```

Donde:

nombremetodo	: Es el nombre del método invocado.
variable	: Es la variable que recibe el valor de retorno.

FUNCIÓN EN PROGRAMACIÓN

Este tipo de método recibe datos de entrada a través de parámetros y retorna un resultado al punto donde es invocado. En la Figura 2, p1 y p2 son parámetros.

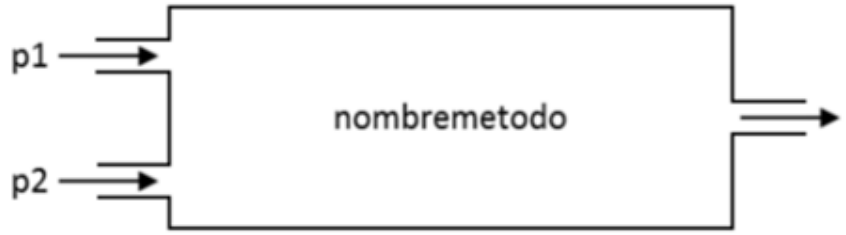


Figura 2 Método con valor de retorno y con parámetros

Este tipo de método se define de la siguiente manera:

```
tiporetorno nombremetodo (tipo1 p1, tipo2 p2, tipo3 p3, . . .) {
        Declaración de variables locales
        Cuerpo del método
        return valor;
}
```

Donde:

`nombremetodo`	: Es el nombre del método.
`tiporetorno`	: Es el tipo del valor de retorno.
`p1, p2, p3, ...`	: Son los nombres de los parámetros.
`tipo1, tipo2, tipo3, ...`	: Son los tipos de los parámetros.
`valor`	: Es el valor de retorno.

Llamada

Este tipo de método se invoca de la siguiente manera:

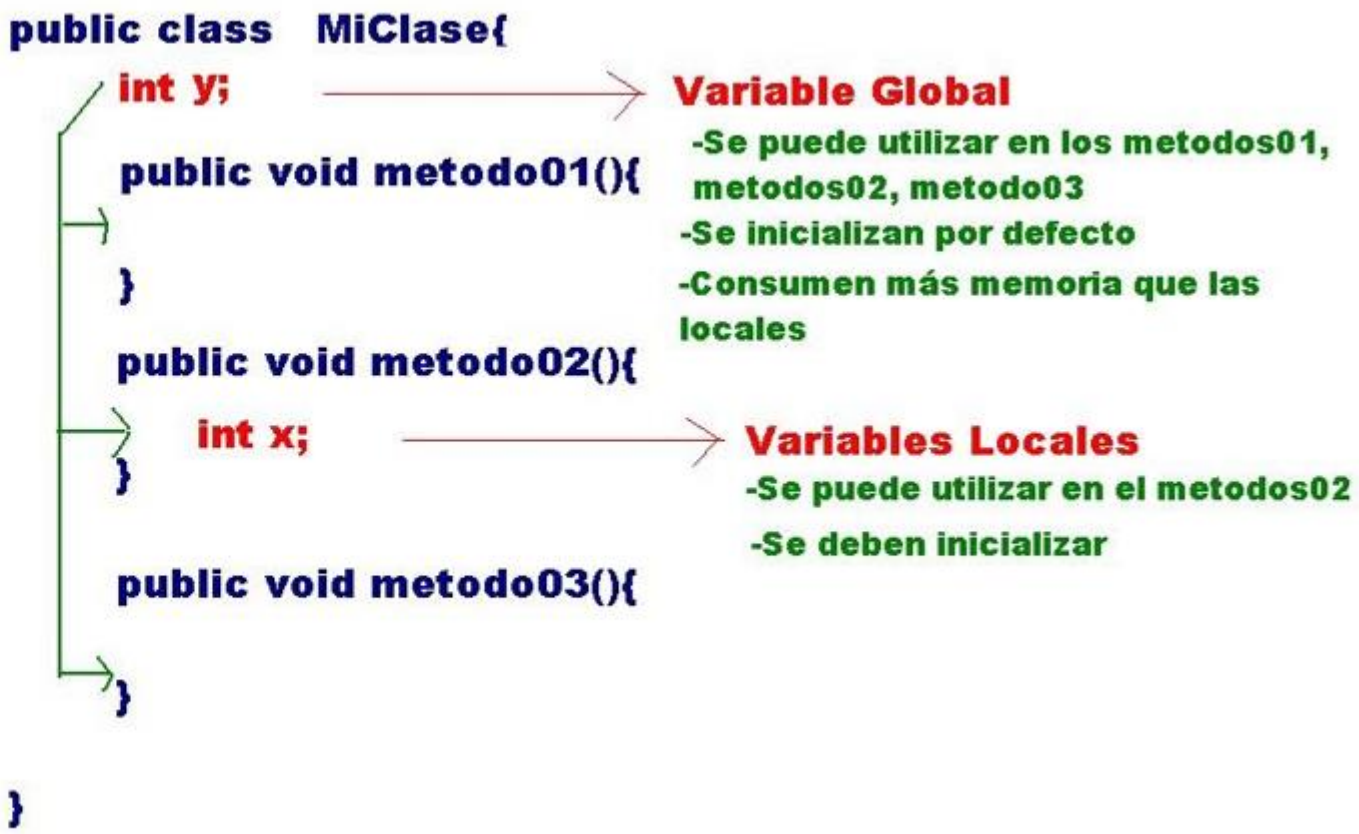
```
variable = nombremetodo (v1, v2, v3, ...);
```

Donde:

`nombremetodo`	: Es el nombre del método invocado.
`variable`	: Es la variable que recibe el valor de retorno.
`v1, v2, v3, ...`	: Son los valores dados a los parámetros.

FUNCIÓN EN PROGRAMACIÓN

Los métodos pueden utilizar sus propias variables denominadas variables locales o variables de uso compartido, comunes a todos los métodos, denominadas variables globales



FUNCIÓN EN PROGRAMACIÓN

Variables locales: Una variable local es una variable que se declara en el interior de un método por lo que su ámbito es el interior del método, es decir, sólo puede ser utilizada dentro del método donde fue declarada. Este tipo de variable se crea vacía al iniciar la ejecución del método y se destruye al finalizar la ejecución del método.

Variables globales: Una variable global es una variable que se declara dentro del programa, pero en el exterior de todos los métodos; por lo que, su ámbito es el interior de todo el programa, es decir, puede ser utilizada en cualquier parte del programa. Este tipo de variable se crea al iniciar la ejecución del programa y se destruye al finalizar. Por otro lado, una variable global se inicializa automáticamente.

FUNCIÓN EN PROGRAMACIÓN

Concepto	Variabe Local	Variable Global
Declaración	En el interior de un método	En el exterior de todos los métodos
Creación	Al iniciar la ejecución del método	Al iniciar la ejecución del programa
Destrucción	Al finalizar la ejecución del método	Al finalizar la ejecución del programa
Alcance	Todo el método	Todo el programa
Inicialización	No se inicializa automáticamente	Se inicializa automáticamente

Tipo	Valor inicial
int	0
double	0.0
String	null

FUNCIÓN EN PROGRAMACIÓN

Un método tipo void es un módulo de programa que puede recibir datos de entrada a través de variables locales denominadas parámetros; pero que no retorna ningún resultado al punto donde es invocado, razón por el que se le conoce también como método sin valor de retorno.

Los métodos tipo void pueden dividirse a su vez en dos tipos:

- Métodos tipo void sin parámetros
- Métodos tipo void con parámetros

FUNCIÓN EN PROGRAMACIÓN

Estos métodos no pueden recibir datos de entrada ni retornar ningún resultado al punto de su invocación.

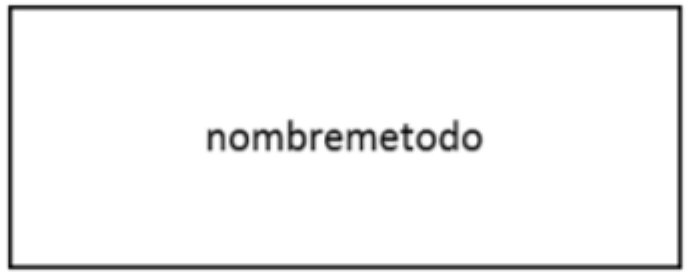


Figura 3 Método tipo void sin parámetros

Cuando se programa usando métodos, se siguen dos etapas. Primero, el método debe definirse. Esto consiste en crear el método ubicándolo en alguna parte del programa. Segundo, el método creado debe ser invocado en el lugar donde se requiera. Esto consiste en poner el método en ejecución.

FUNCIÓN EN PROGRAMACIÓN

Definición

Este tipo de método se define de la siguiente manera:

```
void nombremetodo() {
    Declaración de variables locales
    Cuerpo del método
}
```

Donde:

nombremetodo : Es el nombre del método.

Llamada

Este tipo de método se invoca de la siguiente manera:

```
nombremetodo();
```

Donde:

nombremetodo : Es el nombre del método invocado.

FUNCIÓN EN PROGRAMACIÓN

Estos métodos reciben datos de entrada a través de variables locales al método denominadas parámetros; pero, igual que en el caso anterior, no pueden retornar ningún resultado al punto de su invocación.

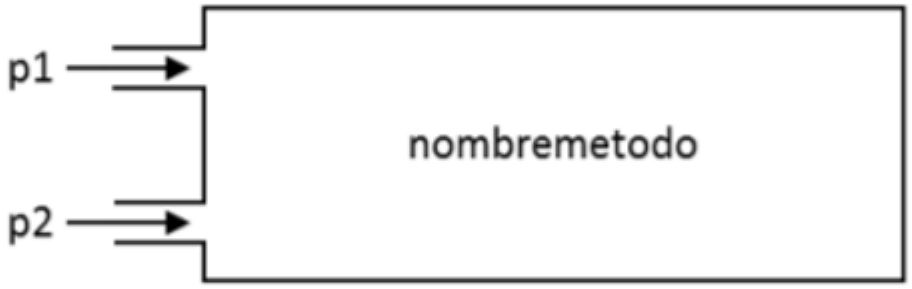


Figura 4 Método tipo void con parámetros

Donde p1, p2, etc son los parámetros del método. El número de parámetros es variable y depende de las necesidades del método.

FUNCIÓN EN PROGRAMACIÓN

Definición

Este tipo de método se define de la siguiente manera:

```
void nombremetodo(tipo1 p1, tipo2 p2, ...) {
    Declaración de variables locales
    Cuerpo del método
}
```

Donde:

`nombremetodo`	: Es el nombre del método.
`p1, p2,...`	: Son los nombres de los parámetros.

Llamada

Este tipo de método se invoca de la siguiente manera:

```
nombremetodo(v1, v2, ...);
```

Donde:

`nombremetodo`	: Es el nombre del método invocado.
`v1, v2,...`	: Son los valores pasados a los parámetros.

FUNCIÓN EN PROGRAMACIÓN

¿PREGUNTAS?

UPN

UNIVERSIDAD
PRIVADA
DEL NORTE