

UPN

**UNIVERSIDAD
PRIVADA
DEL NORTE**

FUNDAMENTOS DE ALGORITMOS



Ing. Yordan Marin Sovero

LOGRO DE LA SESIÓN 8

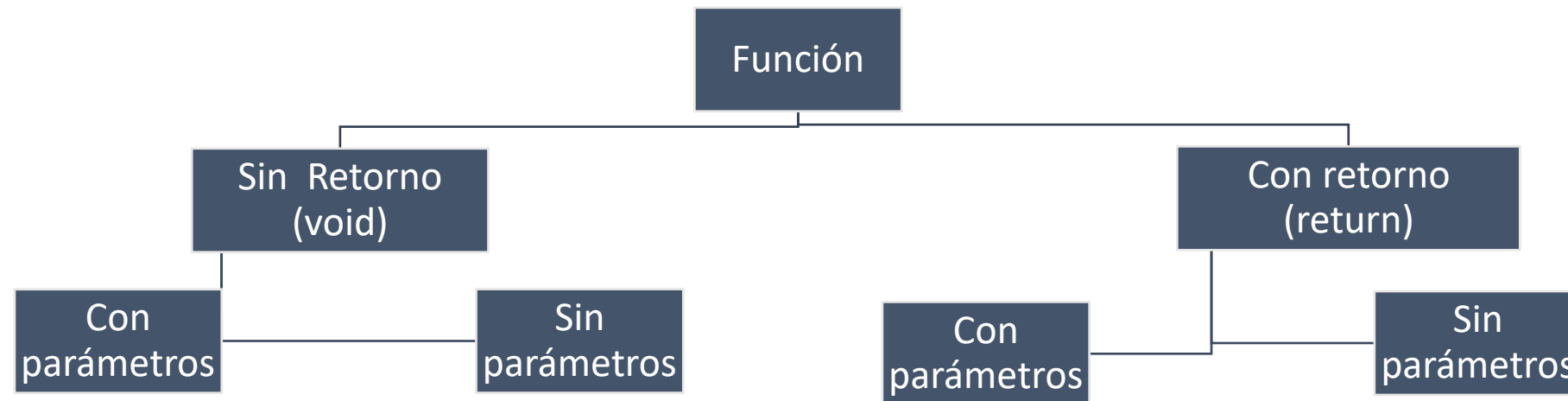
Al finalizar la sesión, el estudiante entiende los tipos de funciones, alcance de variables y parámetros, mostrando dominio técnico en el lenguaje de programación c#.



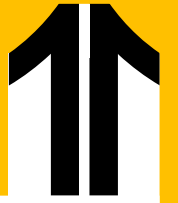
FUNCIÓN



Es un bloque de código que realiza una tarea específica y puede aceptar datos de entrada, llamados parámetros, y opcionalmente devolver un resultado. En la programación, las funciones son una forma de organizar y reutilizar código, ya que permiten encapsular cierta funcionalidad y llamarla desde diferentes partes del programa.



FUNCIÓN SIN RETORNO (void)



Estas funciones realizan una tarea pero no devuelven ningún valor.

SIN RETORNO SIN PARÁMETROS

```
static void Main(string[] args)
{
    Nombre();
}
```

1 referencia

```
static void Nombre()
{
    string nom = "Juan";
    Console.WriteLine(nom);
    Console.ReadKey();
}
```

SIN RETORNO CON PARÁMETROS

```
static void Main(string[] args)
{
    Nombre("Juan");
}
```

1 referencia

```
static void Nombre(string nom)
{
    Console.WriteLine(nom);
    Console.ReadKey();
}
```

FUNCIÓN CON RETORNO (return)



Estas funciones realizan una tarea y devuelven un valor como resultado. Se especifica el tipo de dato que retornan la función (puede ser cualquier tipo de dato válido en C#).

CON RETORNO SIN PARÁMETROS

```
static void Main(string[] args)
{
    Console.WriteLine(Nombre());
    Console.ReadKey();
}
```

1 referencia

```
static string Nombre()
{
    string nombre = "Yordan";
    return nombre;
}
```

CON RETORNO CON PARÁMETROS

```
static void Main(string[] args)
{
    Console.WriteLine(Nombre("Yordan"));
    Console.ReadKey();
}
```

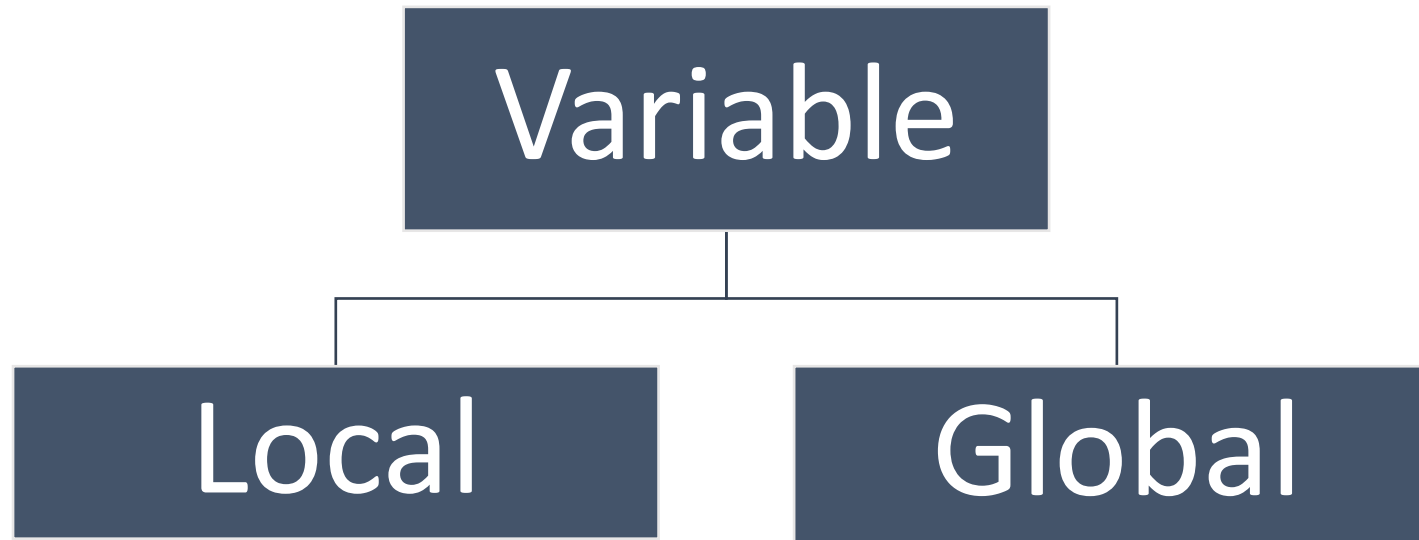
1 referencia

```
static string Nombre(string nombre)
{
    return nombre;
}
```

VARIABLES



Una variable es un espacio de memoria con un nombre asociado que se utiliza para almacenar datos que pueden cambiar durante la ejecución de un programa



VARIABLE LOCAL



- Son aquellas que se declaran dentro de un bloque de código, como una función o un método.
- Solo son visibles y accesibles dentro del bloque en el que fueron declaradas.
- Su vida útil está limitada al tiempo de ejecución del bloque en el que se declararon.
- No se pueden acceder desde fuera del bloque en el que se declararon.

```
static void Main(string[] args)
{
    int edad = 30; // <--- variable local
    Console.WriteLine("Edad: " + edad);
    Console.ReadKey();
}
```


VARIABLE GLOBAL



- Son aquellas que se declaran fuera de cualquier bloque de código, generalmente al inicio de un archivo o clase.
- Son visibles y accesibles desde cualquier lugar del programa.
- Su vida útil está determinada por la ejecución del programa.
- Se pueden acceder y modificar desde cualquier parte del programa, lo que puede hacer que sean propensas a errores si no se utilizan correctamente.

```
static int edad = 30; // <--- variable global
```

0 referencias

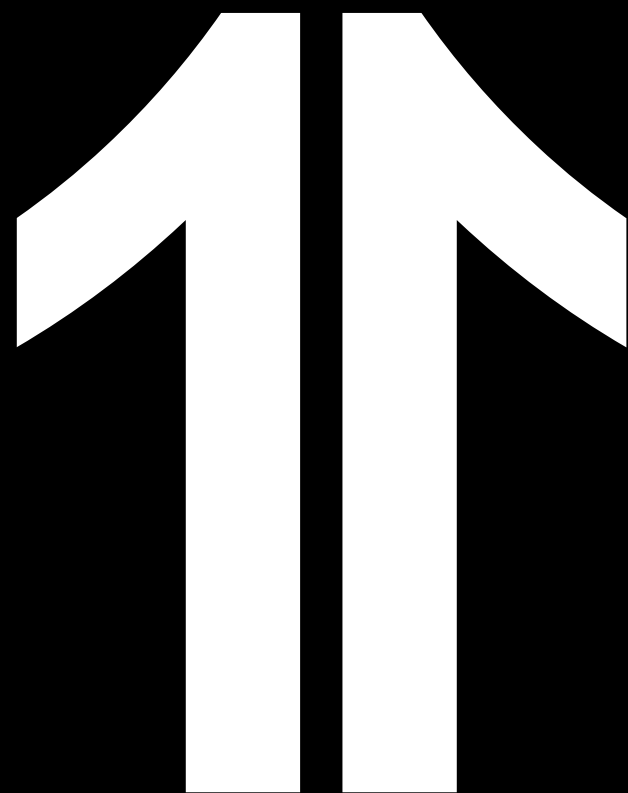
```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("Edad: " + edad);
```

```
    Console.ReadKey();
```

```
}
```



UPN

**UNIVERSIDAD
PRIVADA
DEL NORTE**