| Category | Operators |
|---|---|
| Primary | x.y f(x) a[x] x++ x– new typeof default checked unchecked |
| Unary | + - ! ~ ++x –x (T)x |
| Multiplicative | * / % |
| Additive | + - |
| Shift | << >> |
| Relational and type testing | < > <= >= is as |
| Equality | == != |
| Logical AND | & |
| Logical XOR | ^ |
| Logical OR | \| |
| Conditional AND | && |
| Conditional OR | \|\| |
| Null coalescing | ?? |
| Conditional | ?: |
| Assignment and lambda expression | = *= /= %= += -= <<= >>= &= ^= \|= => |

| Statement | Example |
|---|---|
| Local variable declaration | `static void () {`<br>`  int a;`<br>`  int b = 2, c = 3;`<br>`  a = 1;`<br>`  Console.WriteLine(a + b + c);`<br>`}` |
| Local constant declaration | `static void () {`<br>`  const float pi = 3.1415927f;`<br>`  const int r = 25;`<br>`  Console.WriteLine(pi * r * r);`<br>`}` |
| Expression statement | `static void () {`<br>`  int i;`<br>`  i = 123;  // Expression statement`<br>`  Console.WriteLine(i); // Expression statement`<br>`  i++;  // Expression statement`<br>`  Console.WriteLine(i); // Expression statement`<br>`}` |
| if statement | `static void (string[] args) {`<br>`  if (args.Length == 0) {`<br>`  Console.WriteLine("No arguments");`<br>`  }`<br>`  else {`<br>`  Console.WriteLine("One or more arguments");`<br>`  }`<br>`}` |

```
static void Main()          Literals
{                             ↓
   Console.WriteLine("{0}", 1024);          // int literal
   Console.WriteLine("{0}", 3.1416);        // double literal
   Console.WriteLine("{0}", 3.1416F);       // float literal
   Console.WriteLine("{0}", true);          // boolean literal
   Console.WriteLine("{0}", 'x');           // character literal
   Console.WriteLine("{0}", "Hi there");    // string literal
}
```

The output of this code is the following:

```
1024
3.1416
3.1416
True
x
Hi there
```

```
using System;
using System.Text;

public class EntryPoint
{
    static void Main() {
        StringBuilder sb = new StringBuilder();

        sb.Append("StringBuilder ").Append("is ")
            .Append("very... ");

        string built1 = sb.ToString();

        sb.Append("cool");

        string built2 = sb.ToString();

        Console.WriteLine( built1 );
        Console.WriteLine( built2 );
    }
}
```

```
String[] resultArray = s1.Split( delimiters );
foreach ( String subString in resultArray )
{
    Console.WriteLine(ctr++ + ":" + subString);
}
```

Una clase abstracta no
instanciada, sirve para ha
base para otras clases. S
puede tener construc
declarados con el mismo

int s,b,c,d;
a = b =c =d =e =20;

Estan formadas por operandos y Operadores y evaluan valores.

indican la opetacion a realizar

Una setencia completa que esta formados por una secuencia completa y termina con ;

Son numeros o cadenas escritas en el codigo fuente que representan un valor a un conjunto de valores

es una matriz de caracteres

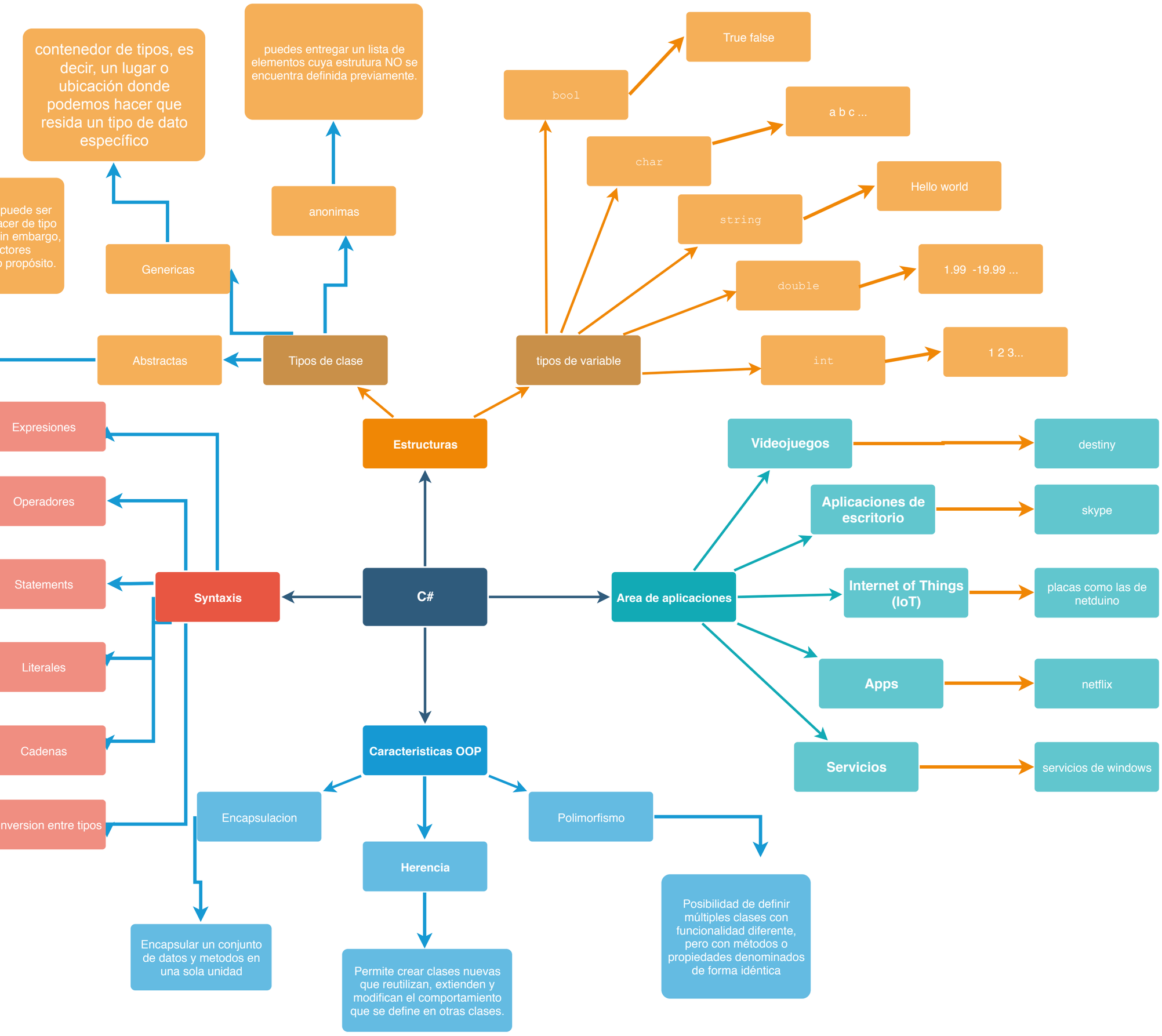
Permite que un tipo sea tratado como otro tipo de dato

Co

**Tipos de Conversión:**

- Explicit Cast
- Implicit Cast
- Without Casting
  - Utilizando el método Parse()
  - Utilizando System.Convert
  - Utilizando ToString()
  - Utilizando tryParse() (nuevo en .NET 2.0)

```
int intNumber = 314
long longNumber =

int intNumber = 314
long longNumber =

string text = "9.11E-
float kgElectronMas

String middleCText
double middleC = S
bool boolean = Sys

bool boolean = true;
string text = boolean.
System.Console.Wri

double number;
string input;
System.Console.Wri
input = System.Cons
if (double.TryParse(i
  // Converted corr
else {
  System.Console.W
number.");
}
```




```
16;
(long) intNumber;
16;
ntNumber;
31";
s = float.Parse(text);
= "278.4375";
System.Convert.ToDouble(middleCText);
tem.Convert.ToBoolean(middleC);
.ToString();
teLine(text);    // Display "True"
te("Enter a number: ");
ole.ReadLine();
nput, out number)) {
ctly, now use number // ... }
/riteLine( "The text entered was not a valid
```