EJERCICIOS CAJA BLANCA

Técnica del camino básico. Ejercicios 5 y 6

Ingeniería del Software DSIC-UPV

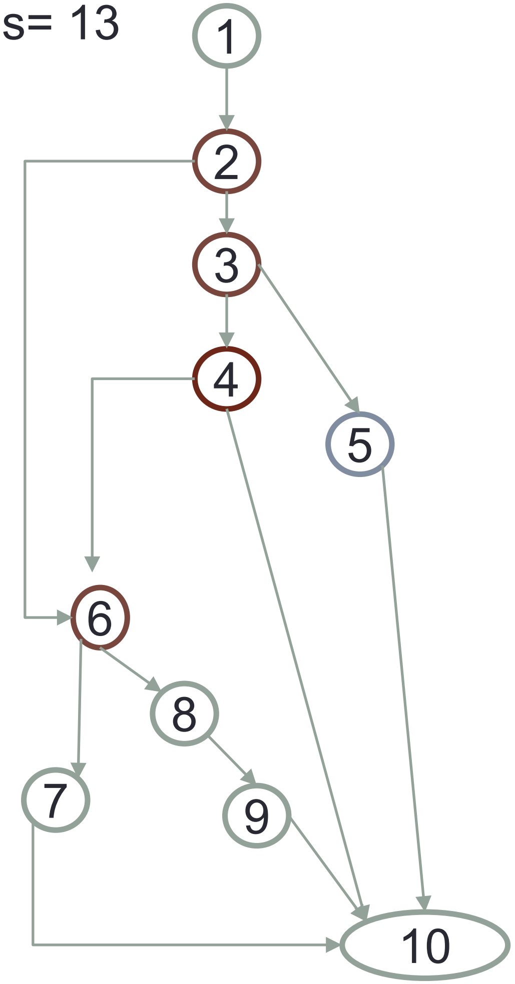
Ejercicio 5

```
public double obtainBonus(string cod_employee, int months)
 {
    int bonus = 0;
    try
    {
       int code = int.Parse(cod_employee);
       if(code > 100)
       {
           bonus = 100 * months;
       }
       else
       {
            bonus = 10;
        }
     }
     catch (FormatException e)
     {
            Console.WriteLine("cod_employee is not a number" +
          e.Message);
     }

     catch (Exception e)
     {
          Console.WriteLine("Other error:" + e.Message);
     }
     return bonus;

  }
```

Ejercicio 5

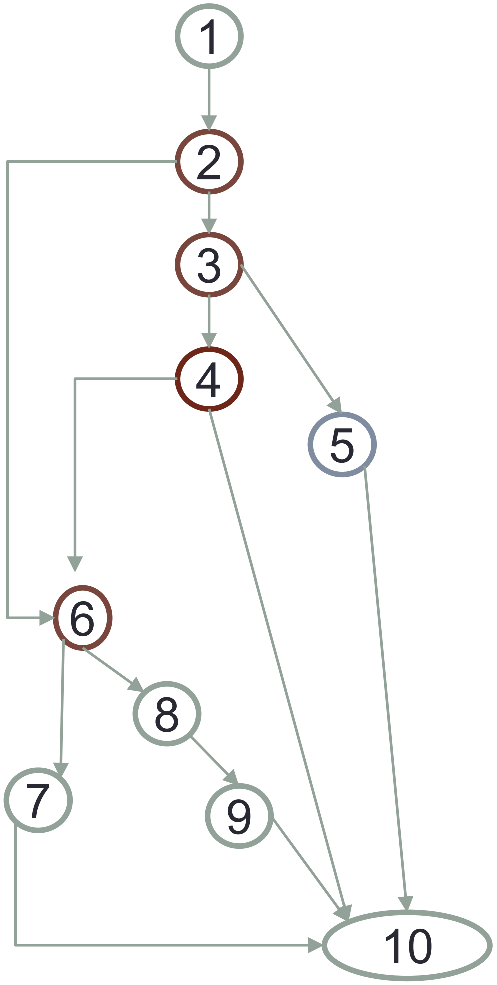
V(G)= 5
Areas = 5
Nodos predicado= 4 → 4+1=5
Nodos= 10 → 13-10+2 =5
Aristas= 13

```
public double obtainBonus(string cod_employee, int months)
 {
    int bonus = 0; 1
    try
    {
       int code = int.Parse(cod_employee); 2
       if(code > 100) 3
       {
          bonus = 100 * months; 4
       }
       else
       {
           bonus = 10; 5
        }
     }
     catch (FormatException e) 6
     {
       7 Console.WriteLine("cod_employee is not a number" +
        e.Message);
     }

     catch (Exception e) 8
     {
     9 Console.WriteLine("Other error:" + e.Message);
     }
     return bonus; 10

  }
```



Ejercicio 5

Camino
{1-2-6-7-10} cod_employee no es número
{1-**2-3**-4-10} Cod_employee>100
{1-2-**3-5**-10} Cod_employee <=100
{1-2-**6-8**-9-10} No posible
{1-2-3-**4-6**-8-9-10} Operación con excepción



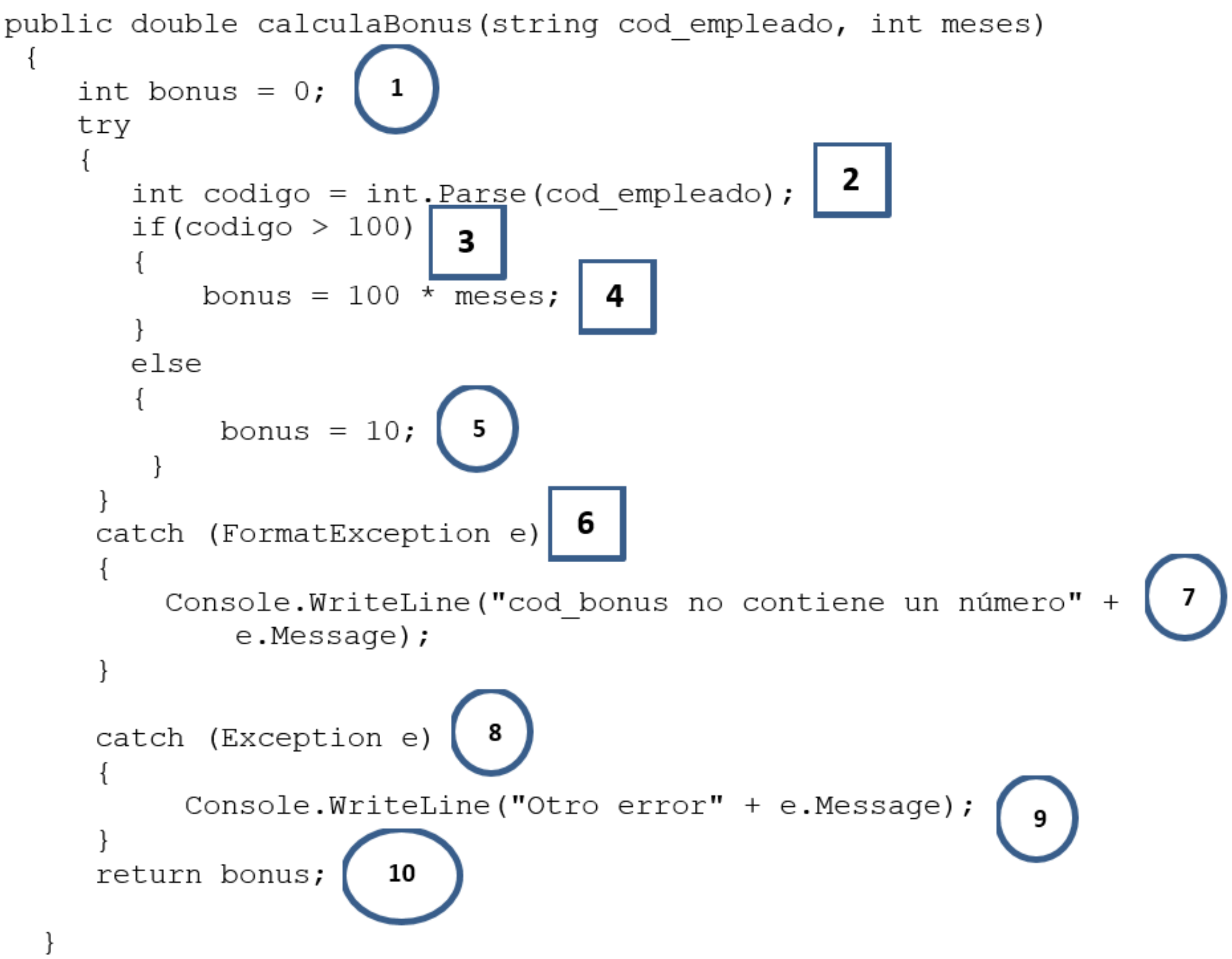
Ejercicio 5

Camino	Entrada		Salida
,	**Cod_employee**	**Months**	
{1-2-6-7-10} cod_employee no es número	“A”	2	Cod_employee is not a number 0
{1-**2-3**-4-10} Cod_employee>100	“1000”	2	200
{1-2-**3-5**-10} Cod_employee <=100	“90”	2	10
{1-2-**6-8**-9-10} No posible	No posible		No posible
{1-2-3-**4-6**-8-9-10} Operación con excepción	“1000”	Int.MaxValue	Other error 0

Ejercicio 6

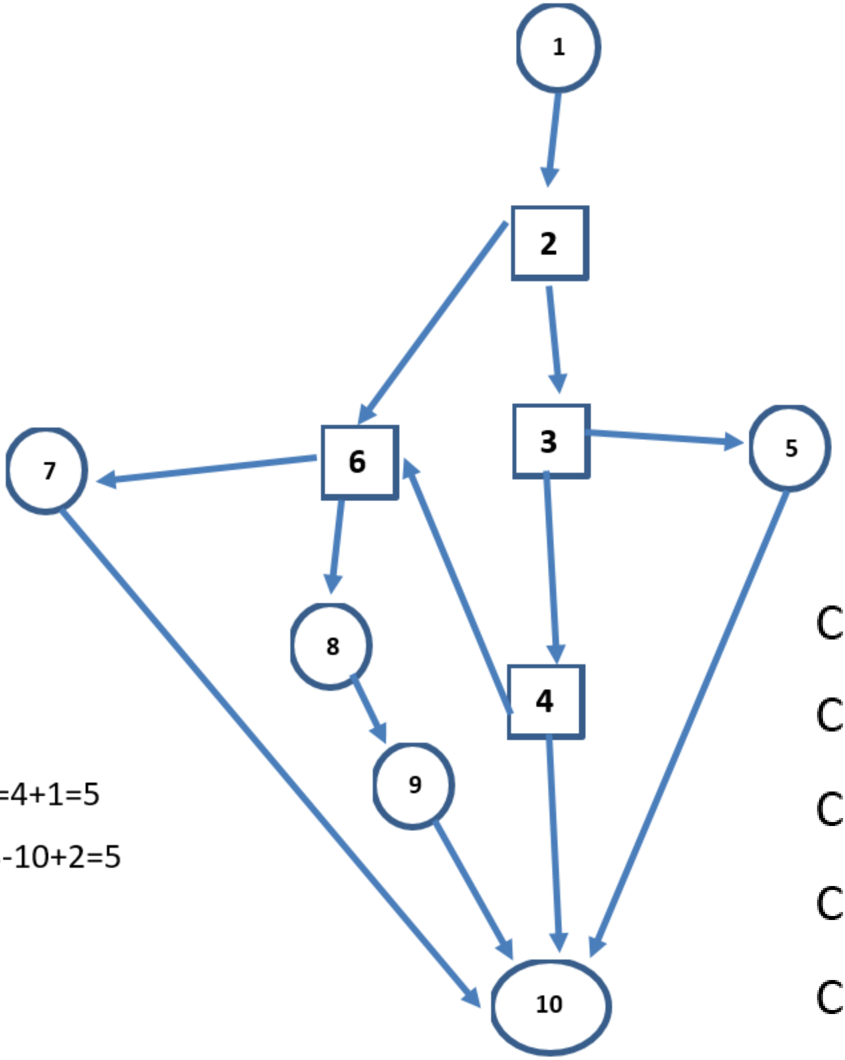
```
public double calculaBonus(string cod_empleado, int meses)
{
   int bonus = 0;
   try
   {
     int codigo = int.Parse(cod_empleado);
     if(codigo > 100)
     {
        bonus = 100 * meses;
     }
     else
     {
         bonus = 10;
      }
    }
    catch (FormatException e)
    {
                        Console.WriteLine("cod_bonus no contiene un número" +
e.Message);
    }

    catch (Exception e)
    {
        Console.WriteLine("Otro error" + e.Message);
    }
    return bonus;

 }
```

Ejercicio 6



```
public double calculaBonus(string cod_empleado, int meses)
 {
    int bonus = 0;
    try
    {
       int codigo = int.Parse(cod_empleado);
       if(codigo > 100)
       {
           bonus = 100 * meses;
       }
       else
       {
            bonus = 10;
        }
     }
     catch (FormatException e)
     {
        Console.WriteLine("cod_bonus no contiene un número" +
            e.Message);
     }

     catch (Exception e)
     {
         Console.WriteLine("Otro error" + e.Message);
     }
     return bonus;

  }
```

Ejercicio 6



CC = Nº Regiones = 5

CC=Nodos Predicado+1=4+1=5

CC=Aristas-Nodos+2=13-10+2=5

Camino 1: 1-2-6-7-10

Camino 2: 1-2-3-4-10

Camino 3: 1-2-3-5-10

Camino 4: 1-2-6-8-9-10

Camino 5: 1-2-3-4-6-8-9-10

Ejercicio 6

Camino	Entrada		Salida
	Cod_empleado	meses	
1	“A”	2	Cod_bonus no contiene un número 0
2	“1000”	2	200
3	“90”	2	10
4			No hay entradas que permitan probar este camino
5	“1000”	Int.MaxValue	Otro error 0